НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА АВТОМАТИКИ ТА УПРАВЛІННЯ В ТЕХНІЧНИХ СИСТЕМАХ

«На правах рукопису»
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ _____
(підпис) (ініціали, прізвище)
“___”_____20__ р.

Магістерська дисертація

зі спеціальності (спеціалізації) 121, інженерія програмного забезпечення (інженерія програмного забезпечення комп’ютерних систем)
(код і назва спеціальності)

на тему: Програмна система економічного аналізу підприємств на основі теорії ігор

Виконав (-ла): студент (-ка) 2 курсу, групи ІТ-83мп
(шифр групи)

Заверткін Іван Анатолійович
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник професор, д.ф.-м.н., проф., Дорошенко А.Ю.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет (інститут) Інформатики та обчислювальної техніки
(повна назва)

Кафедра Автоматики та управління в технічних системах
(повна назва)

Освітньо-кваліфікаційний ступінь магістр
(назва ОКР)

Спеціальність 121 Інженерія програмного забезпечення
(код і назва)

Спеціалізація 121 Інженерія програмного забезпечення комп'ютерних систем
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
О.І. Ролік
(підпис) (ініціали, прізвище)
« » 2019 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Заверткін Іван Анатолійович
(прізвище, ім'я, по батькові)

1. Тема дисертації Програмна система економічного аналізу підприємств на основі теорії ігор
Науковий керівник дисертації Дорошенко А. Ю. д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «____» _____ 2019 р. № _____

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження система економічного аналізу підприємств
_____

4. Зміст пояснювальної записки а) аналіз математичної моделі; б) огляд існуючих рішень; в) опис архітектурної реалізації та засобів розробки; г) опис програмної реалізації; д) інтерфейс користувача; е) тестування; є) розробка стартап-проекту .

5. Перелік завдань, які потрібно розробити: проаналізувати предметну область, проаналізувати існуючі рішення, розробити архітектуру проекту, розробити програмну реалізацію, розробити інтерфейс користувача, провести тестування, провести маркетинговий аналіз стартап проекту.

6. Перелік графічного (ілюстративного) матеріалу: діаграма діяльності, діграма розгортання, діаграма прецедентів, діаграма класів, діаграма пакетів front-end, діаграма пакетів server, діаграма послідовності процесу використання, діаграма послідовності роботи сервера.

7. Орієнтовний перелік публікацій: _____

_____

8. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 04 вересня 2019

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів дисертації	Примітка
1	Аналіз математичної моделі	15.09.2019	
2	Огляд існуючих рішень	18.09.2019	
3	Розробка архітектури реалізації	25.09.2019	
4	Розробка програмної реалізації	20.10.2019	
5	Розробка інтерфейсу користвача	25.10.2019	
6	Тестування та аналіз роботи системи	10.11.2019	
7	Розробка стартап-проекту	20.11.2019	
8	Оформлення матеріалів дисертації	04.12.2019	

Студент _____ І. А. Заверткін
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ А. Ю. Дорошенко
(підпис) (ініціали, прізвище)

РЕФЕРАТ

Магістерська дисертація на здобуття ступеня магістру на тему “Програмна система економічного аналізу підприємств на основі теорії ігор”: 101с., 19 рис., 30 табл., 1 додатки, 25 джерел.

Об’єкт дослідження - система економічного аналізу підприємств.

Мета роботи – розробка веб-додатку, для аналізу стратегій за теорією ігор.

У зв'язку з підвищенням автоматизації процесів на підприємстві все більшої актуальності набувають системи покращення результатів виробництва. Таким чином дана система є ідеальним інструментом для аналізу та формування рішень.

У магістерській дисертації розглянуто основні підходи до аналізу стратегій поведінки на ринку. Пропонується підхід для більш точного аналізу стратегії на основі теорії ігор. Підхід базується на визначенні найкращого результату під час аналізу ринку за допомогою математичних критеріїв і вибору найпривабливішого результату. Важливою особливістю є можливість порівняння усіх найкращих випадків аналізу, що були проаналізовані раніше.

Ключові слова: теорія ігор, веб-додаток, аналіз, стратегія, критерій, математичний аналіз, результат, підприємство.

ABSTRACT

Aster's thesis for master's degree on the topic "Software system of economic analysis of enterprises based on game theory ": 101p, 19 figures, 30 tables, 1 annexes, 25 sources.

Object of study - system of economic analysis of enterprises..

The purpose of the work - developing a web application for economic analyzing on bases game theory.

Due to the increase of process automation, systems of improving production results are becoming increasingly relevant. Thus, this system is an ideal tool for analysis and decision making.

In the master's thesis the basic approaches to the analysis of strategies of behavior on the market are considered. An approach is proposed for a more accurate analysis of strategy based on game theory. The approach is based on determining the best result when analyzing the market using mathematical criteria and selecting the most attractive result. An important feature is the ability to compare all the best analysis cases that have been analyzed previously.

Keywords: game theory, web application, analysis, strategy, criterion, mathematical analysis, result, enterprise.

ЗМІСТ

ПЕРЕЛІК ВИКОРИСТАНИХ У РОБОТІ СКОРОЧЕНЬ 8
ВСТУП 9
1 АНАЛІЗ МАТЕМАТИЧХ МОДЕЛЕЙ 11
1.1 Теорія ігор як метод економічного аналізу 11
1.2 Основні поняття теорії ігор. Класифікація ігор 16
1.3 Вирішення ігор в чистих та змішаних стратегіях 19
1.4 Графічне вирішення ігор лінійного програмування 27
1.5 Висновок 31
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ 32
2.1 Поняття CRM системи 32
2.2 Опис існуючих рішень 37
2.3 Висновок 49
3 ОПИС АРХІТЕКТУРНОЇ РЕАЛІЗАЦІЇ ТА ЗАСОБІВ РОЗРОБКИ 51
3.1 Компонентний підхід ..... **Ошибка! Закладка не определена.**
3.2 Допоміжні компоненти в реалізації fronted частини 53
3.3 Висновок 56
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ 57
4.1 Опис основної архітектури додатку 57
4.2 Основні класи 57
4.2.1 Клас HomeComponent 57
4.2.2 Клас ConditionsCompoent 58
4.2.3 Клас BayesCriteria 58
4.2.5 Клас HurwitzCriteria 59
4.2.6 Клас LaplaceCriteria 60
4.2.7 Клас OptimismCriteria 60
4.2.8 Клас SavageCriteria 60
4.2.9 Клас WaldCriteria 61
4.3 Висновок 61

5. ІНТЕРФЕЙС КОРИСТУВАЧА 62
5.1 Основні частини інтерфейсу 62
5.2 Висновок 67
6. ТЕСТУВАННЯ ..... **Ошибка! Закладка не определена.**
6.1 Компонентне тестування ..... **Ошибка! Закладка не определена.**
6.2 Інтеграційне тестування ..... **Ошибка! Закладка не определена.**
6.3 Системне тестування ..... **Ошибка! Закладка не определена.**
6.4 Приймальне тестування 72
6.5 Результати тестування 72
6.6 Висновок 74
7 СТАРТАП-ПРОЕКТ 76
7.1 Опис ідеї 76
7.2 Технологічний аудит проекту 77
7.3 Аналіз ринкових можливостей запуску стартап-проекту 78
7.5 Розроблення маркетингової програми стартап-проекту 87
7.6 Висновки 90
ВИСНОВКИ 92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 93
ДОДАТКИ 95
Додаток А. Лістинг коду системи 95

ПЕРЕЛІК ВИКОРИСТАНИХ У РОБОТІ СКОРОЧЕНЬ

SPA – Single Page Application

API – Application Programming Interface

URL – Uniform Resource Locator

UX – User Experience

UI – User Interface

CRM - Customer Relationship Management

MS - Microsoft

JS - JavaScript

CSS - Cascading Style Sheets

RxJS – Reactive Extensions Library for JavaScript

ПЗ – Програмне забезпечення

ВСТУП

Ринкові відносини припускають наявність конкурентного середовища, де кожен учасник економіки прагне до досягнення власних цілей і інтересів. Виживання на ринку, підтримання тривалої конкурентоспроможності багато в чому залежить від уміння аналізувати власні слабкі і сильні сторони, а також прогнозувати поведінку основних конкурентів.

Досить часто для визначення найбільш ефективної моделі поведінки на ринку використовується теорія ігор. Саме таке широке використання теорії ігор в програмному економічному аналізі визначає актуальність обраної тематики.

Ринкова економіка в ідеалі повинна формувати відкриту і прозору інформацію, проте, в практичному житті через конкурентну боротьбу це неможливо.

Теорія ігор – це розділ математичної економіки, що вивчає рішення конфліктів між гравцями і оптимальність їх стратегій.

Конфлікт може відноситись до різних областей людського інтересу: найчастіше це економіка, соціологія, політологія, рідше біологія, кібернетика і навіть військова справа. Конфліктом є будь-яка ситуація, в якій порушені інтересу двох і більше учасників, яких традиційно називають гравцями. Для кожного гравця існує певний набір стратегій, які він може застосувати. Перетинаючись, стратегії декількох гравців створюють певну ситуацію, в якій кожен гравець отримує певний результат, званий виграшем, позитивним або негативним. При виборі стратегії важливо враховувати не тільки отримання максимального профіту, але так само можливі кроки супротивника, і їх вплив на ситуацію в цілому.

У теорії ігор розглядаються варіанти з повною і неповною інформацією. При повній всі суб’єкти мають дані про всіх скоєні раніше кроки своїх конкурентів. Більшість математичних моделей будується на уявленні про неповне володіння інформацією. В економіці, як правило, розглядаються гри з кінцевим числом ходів, на відміну від математичного підходу.

Об’єктом дослідження є теорія ігор як метод програмного економічного аналізу, предметом – використання теорії ігор в розробці програмної системи.

Метою магістерської дисертації є розробка програмної системи економічного аналізу на основі теорії ігор.

Досягнення такої мети потребує виконання наступних завдань:

- розгляду поняття теорії ігор;
- дослідження класифікації ігор;
- розгляду вирішення ігор в чистих та змішаних стратегіях та графічне вирішення ігор;
- проведенню огляду існуючих рішень, опису архітектурної реалізації та засобів розробки;
- опису програмної реалізацію та процес впровадження програмного забезпечення;
- проведенню тестування.

В роботі використано такі методи як аналітичний, метод пошуку рішень, дослідницький.

1 АНАЛІЗ МАТЕМАТИЧХ МОДЕЛЕЙ

1.1 Теорія ігор як метод економічного аналізу [1]

Окремі завдання на пошук оптимальних рішень в конфліктних ситуаціях ставилися математиками ще в XVIII столітті. Наприклад, завдання виробництва і ціноутворення в умовах олігополії, що стала згодом хрестоматійним прикладом у теорії ігор, була поставлена і вирішена французьким математиком А. Курно в XIX столітті. На початку XX століття Е. Ласкер, Е. Цермело, Е. Борель висунули ідею математичної теорії конфлікту інтересів.

Математична теорія ігор бере свою початок з неокласичної економічної теорії. На початку XX століття було усвідомлено, що в економічній теорії відсутня важлива частина – теорія, що описує прийняття рішень учасниками ринку. Два видатних математика - економіста свого часу Оскар Моргенштерн і Джон фон Нейман поставили собі за мету знайти відповідь на це питання. Дослідники прийшли до висновку, що поведінка учасників ринку найбільше схожа на поведінку гравців, що змагаються між собою. За результатами своїх досліджень в 1944р. Монгерштейн і фон Нейман опублікували книгу «Теорія ігор і економічна поведінка », в якій сформулювали визначення гри як діяльності двох або більше учасників, які прагнуть до певних виграшів, здатних розпоряджатися будь-якими ресурсами, взаємодіяти між собою і приймати рішення, засновані на аналізі поведінки інших учасників. Крім того, в книзі математично описаний спосіб пошуку оптимальних рішень учасників гри. Відзначимо, що в монографії Монгерштерна і фон Неймана розглядалися переважно гри з нульовою сумою (коли сума виграшів всіх учасників дорівнює 0, тобто виграш одних – це програш інших) і кооперативні (або коаліційні) ігри, тобто ігри, в яких гравці можуть вступати в вигідні їм коаліції, укладаючи взаємо обов’язкові угоди.

Подальший розвиток теорія ігор отримала завдяки працям Джона Неша. Через 5 років після публікації книги «Теорія ігор і економічна поведінка», в 1949 р. він написав дисертацію по теорії ігор, присвячену безкоаліційним іграм, в яких загальний виграш гравців не дорівнює нулю при будь-якому результаті гри (ігри з

ненульовою сумою; в яких крім протилежних у гравців є і спільні інтереси). Центральною ідеєю Неша є концепція рівноваги, нині носить його ім'я. Рівновага по Нешу – це така комбінація стратегій учасників конфлікту, при якій жоден з учасників не зацікавлений в односторонньому порядку змінювати свою стратегію. Ця концепція отримала назву «Рівновага Неша», яка стало типовим інструментом аналізу майже у всіх розділах економічної теорії, коли необхідний комплексний аналіз взаємодії економічних суб'єктів. Концепція Неша активно застосовується в аналізі конкуренції, олігополії, теорії промислової організації, в макроекономіці при аналізі економічної політики, охорони овкілля. В економіці інформації Неш запропонував базове рішення по операціях для ігор як з фіксованими, так і зі змінними погрозами. Роботи Неша заклали основи для теорії кооперативних і некооперативних ігор як самостійної теоретичної дисципліни.

У 1960-і рр. в працях математика Ізраеля Роберта Джона Аумана стала розвиватися теорія повторюваних ігор, що представляє собою модель взаємодії учасників, повторювана багато разів. В рамках цієї теорії вдається пояснити безліч феноменів, зокрема, наприклад, чому спільна робота ускладнюється при великій кількості учасників, або чому вони рідко взаємодіють, коли висока ймовірність того, що взаємодія припиниться по екзогенних причинах. Схема повторюваних ігор проливає світло на існування і функціонування різних громадських інститутів: від торгових гільдій до Світової організації торгівлі і мафії.

У 1960 році американський економіст Томас Шеллінг опублікував книгу «Стратегія конфлікту», в якій він обґрунтував, що формальний опис гри як набору гравців, стратегій і виграшів недостатньо для опису того, що гравці приймають рішення в реальних конфліктних ситуаціях. В цьому сенсі показовий наступний приклад, відомий як "завдання про зустріч".

Двоє людей неодмінно повинні зустрітися в одному місті завтра о 12:00, і, хоча вони знають дату і час, у них немає ніякої можливості домовитися про місце. Зрозуміло, що варіантів рішень у гравців сотні і тисячі – всі помітні місця в місті, де вони обидва можуть одночасно опинитись і отримати позитивний виграш в тому і тільки в тому випадку, якщо вони опиняться одночасно на одному місці, і нулі –

якщо вони розминулися. Якщо підійти до даної задачі з формальної точки зору, то вирішити її, ймовірно, немає ніяких шансів. Шеллінг зауважив, що це міркування правильно тільки з формальної точки зору: в реальності у таких двох осіб є відмінний від нуля шанс перетнутися в одному місці, при тому що це має бути місце, яке обом видається найприроднішим. Воно, звичайно, буде залежати від контексту: для Нью-Йорка Шеллінг запропонував Центральний вокзал, два гостя Москви найімовірніше підуть на Червону площу, а якщо десь в місті розминулися чоловік і дружина, то їм найприродніше прийти до себе додому. Такі рівноваги, які в описі гри формально ніяк не відрізняються, проте з точки зору реальних гравців більш вірогідні, ніж інші, Шеллінг назвав фокальними точками.

Людська здатність вибирати такі фокальні точки доведена експериментально і є, мабуть, одним з основних факторів успішної координації широкого кола соціальних взаємодій. В одному з експериментів Шеллінга люди повинні були вибрати "орел" чи "решка", не знаючи, який вибір зробили всі інші, причому виграш кожного з учасників залежав від того, скільки учасників зробили той же самий вибір, що і вони. Якщо міркувати формально, то очікувана кількість збігів для кожного учасника експерименту має дорівнювати 20 або 21. Насправді ж 36 чоловік з 42 вибрали "орел", так що збігів виявилося на 15 більше їх "прогнозованого" числа, - і це при тому, що ймовірність випадкового випадання 36 "орлів" з 42 спроб менше 0,0001. В даному випадку роль координуючого механізму грають, ймовірно, порядок перерахування альтернатив і мовне кліше "орел чи решка?", в якому на першому (більш помітному) місці стоїть варіант "орел". В інших випадках завдання координації складніше і менш однозначне, проте навіть коли групам учасників пропонували, не змовляючись, вибрати одне натуральне число (з нескінченної безлічі можливих!), то близько 40% гравців впоралися з цим завданням, зупинившись на таких числах, як 1, 3, 7 або 13.

Шеллінг був, ймовірно, одним з перших, хто помітив, що раціональна поведінка в конфліктній ситуації може полягати не тільки в тому, щоб максимізувати власний очікуваний виграш, але і також в тому, щоб переконати опонента, якій стратегії гравець буде слідувати. Інакше кажучи, раціональна

поведінку в грі має враховувати, що взаємодія носить довготривалий і багато крокоий характер. Тому іноді можна погіршити своє становище на певному етапі конфлікту заради того, щоб опонент повірив в те, що ви будете дотримуватися певної стратегії в більш довгостроковій перспективі. Приклади рішень такого роду: «щоб довести, що я для вас безпечний, я кладу свій пістолет на землю »; «Щоб переконати вас, що я ні за що не здамся, мені доведеться прикувати себе ланцюгами до цього місця ».

Ці приклади висловлюють суть стратегій, які характеризуються властивостями "достовірних зобов'язань": якщо ви змогли переконати опонента в грі, що будете слідувати якоїсь конкретної стратегії, то він стане виходити з цього як з даності, що обмежить свободу його маневру. Саме така логіка лягла в основу поняття рівноваги, введеного Зельтеном, проте саме Шеллінг першим вказав на важливість даного принципу в стратегічних взаємодіях.

В кінці 1960-х рр. Джон Харсані ввів поняття ігор з неповною інформацією і розробив концепцію Байесових рівноваг. Він розглядав ситуації, коли у одного гравця немає інформації про можливі виграші іншого гравця, і тому він змушений оцінювати їх за можливістю. У серії робіт «Гра з неповною інформацією » розроблена методика аналізу конкретних економічних ситуацій, що виникають у зв'язку з прийняттям рішень в умовній неповної інформації про стан іншого учасника гри. Джон Харсані довів, що для кожної гри з неповною інформацією є еквівалентна гра з повною інформацією.

За допомогою математичного апарату теорії ігор Джон Харсані перетворював ігри з неповною інформацією в ігри з досконалою інформацією. Його роботи заклали основи для економіки інформації – розділу економічної теорії, що вивчає економічні аспекти інформації, як реалізується на ринку в вигляді інформаційних продуктів і послуг та циркулюючої всередині сучасної організації.

Гра з неповною інформацією є зручною моделлю для дослідження конфліктних ситуацій в області міжнародних відносин. Якщо одна зі сторін має інформацію, невідому іншій стороні, то існує три можливі стратегії щодо того, як розпорядитися цією інформацією: приховати її від іншого гравця; передати іншому

гравцеві всю інформацію або її частину; передати іншому гравцеві невірну інформацію, тобто дезінформувати його. Можна навести чимало прикладів застосування цих стратегій: наприклад, від противника зазвичай приховують наступ. У той же час, іноді вигідніше дати противнику знати про свої можливості, щоб уникнути його нападу.

Райнхард Зельтен розширив сферу використання концепції рівноваги Неша в аналізі некооперативних ігор, що мають більше однієї рівноваги. Його головна ідея полягала в застосуванні більш строгих умов гри для того, щоб не тільки зменшити число можливих рівноваг, а й не допустити рівноваг, недоцільних в економічних відносинах. Як уже згадувалося раніше, Зельтен ввів новий важливий принцип оптимальності в теорії ігор – рівновагу.

Його зміст полягає в тому, що дії сторін в деякій конфліктної ситуації будуть однакові, незалежно від того, розігрується вона окремо або є частиною більш загальної гри. Рівновага дозволяє відсіяти рівноваги Неша, засновані на недостовірних загроз для гравців.

Загальним методом визначення скоєних рівноваг поза грою є зворотна індукція, при якій оптимізація ходів гравців починається з кінця гри.

Концепція рівноваги ЗЕЛТ оцінюється як фундаментальне поліпшення рівноваги Неша і широко використовується в аналізі олігополії. Крім того, в роботі «Новий розгляд концепції завершення для пунктів рівноваги в екстенсивних іграх» Зелтен ввів поняття рівноваги «тремтячої руки», що володіє додатковою властивістю стійкості до досить малих відхилень гравців від рівноважних стратегій.

Відзначимо, що практично всі основоположники теорії ігор були співробітниками РЕНД Корпорейшн (Research and Development Corporation) – мозкового центру, створеного під егідою ВПС США в Санта-Моніці (штат Каліфорнія) для досліджень в сфері міжконтинентальних балістичних ракет.

За свої роботи в області теорії ігор Джон Неш, Джон Харсані і Райнхард Зельтен отримали в 1994 р. Нобелівську премію з економіки з обґрунтуванням «за аналіз рівноваги в теорії некооперативної ігор». Також Нобелівська премія з економіки була присуджена Ісраель Роберту Джону Ауману і Томасу Шеллінг за

«збагачення нашого розуміння природи конфліктів і співпраці за допомогою апарату теорії ігор». Роботи лауреатів 1994р. створили насамперед формальний апарат і критерії, що дозволяють визначити "раціональні" результати в статичних і динамічних іграх. Починаючи з 1980-х років цей інструментарій став широко застосовуватися для аналізу різноманітних соціально-економічних взаємодій – від аукціонних торгів до політичних процесів, від теорії міжнародної торгівлі до конфліктів на ринку праці. Список подібних додатків множиться з кожним днем, і тепер, мабуть, вже важко уявити собі який-небудь розділ економічної науки, здатний обійтися без теорії ігор. Ця "експансія" грунтувалася в першу чергу на трьох класичних концепціях: рівноваги Неша для некооперативних ігор; рівноваги, вчиненої поза грою для динамічних ігор з повною інформацією; байесовських рівноваг для ігор з неповною інформацією (тобто на поняттях, введених в літературу Нешем, Зельтеном і Харшані).

Однак ці концепції рівноваг самі по собі не можуть ні вважатися останнім словом в теорії ігор, ні служити інструментом для інтерпретацій змістовних соціальних взаємодій. Так, в переважній більшості ігор, що представляють економічний інтерес, виявляється більше однієї рівноваги Неша, і далеко не всі вони "відсікаються" такими посиленнями рівноваги, як досконалість поза грою і байєсовські рівноваги. Крім того, в останнє десятиліття економісти стали все активніше цікавитися поведінковими і психологічними детермінантами соціальної поведінки, для вивчення яких часто використовуються експериментальні методи.

Під час цих досліджень активно накопичуються знання про реальні взаємодіях живих людей, що не тільки відкриває нові горизонти для теорії ігор, але і збагачує уявлення про природу і характер самої людини.

1.2 Основні поняття теорії ігор. Класифікація ігор [2]

Головним поняття в теорії ігор є протистояння декількох гравців, в якому кожен з них має свій інтерес, даний процес і називається грою. Під час кожної гри можуть виникати конфлікти, які можуть базуватися не тільки на протистоянні гравців, а й на

розбіжностях в погляді на гру. Головною проблемою під час гри може бути поява антагоністичного протистояння, дане протистояння пов’язане з програшною або виграшною ситуацією для однієї з сторін. Також слід зазначити, що антагоністичне протистояння з’являється тільки тоді, коли одна з сторін починає домінувати на певну величину, а інша програє на таку саму кількість. Але якщо інтереси кожної з сторін починають збігатися, протистояння закінчується і починається кооперація.

У більшій кількості ігор, що пов’язані з економічними або управлінськими ситуаціями, протистояння гравців не завжди має антагоністичну природу.

Теорія ігор – це математична теорія для вирішення спірних ситуацій в житі або на ринку.

Мета теорії ігор – це знаходження найліпшого для кожного гравця результату , при увазі на інших учасників гри.

Від реальних протистоянь гра відрізняється тим, що в ній надається увага на правила гри. Основною ідеєю привал є те, що кожна сторона конфлікту має деякий обсяг інформації про кожну з сторін та знає про результати кожного гравця. Також правили встановлюють, що гра буде закінчена коли якась кількість ходів вже зроблена, а більше ходів робити заборонено.

Однак, теорія ігор має деякі обмеження, а саме, що кожний з учасників має повне розуміння про супротивника. Даним правилом користуються, що зрозуміти в чому супротивник може програти і скористатися цим в свою користь.

Ще одним з недоліків теорії, є те гравці повинні знати про кожний майбутній хід супротивника, але вони не можуть знати коли супротивник скористається ходом в партії. На жаль у житі відбувається не так, оскільки перелік ходів супротивника невідомо та найліпше рішення це вихід з гри.

В теорії ігор повністю відсутній елемент ризику. Дана математична модель визначає найкращій та найбезпечніший порядок дій під час конфлікту, гри.

Для визначення оптимального плану на гру використовують певні критерії. Одна практика показу, що орієнтуватися тільки на один з критеріїв неможливо, оскільки при однакових вхідних даних різні критерії можуть показати різні результати це означає, що гравець може обрати неоптимальну стратегію.

Спираючись на ці обмеження та не дотримуючись лише рекомендованих стратегій, що надає теорія ігор, все ж можливо вибрати або створити найбільш зручний шлях для виходу з конфліктної ситуації.

Ще одним критерієм в теорії ігор, є те, що кожен гравець має ходити почергово. Самі ходи поділяються на два типи, випадкові та особисті. Особистим ходом називають тільки тоді, коли гравець свідомо робить хід, що складається з сукупності дій. Випадковим ходом називають, коли за гравця хід робить якийсь механізм або алгоритм. Сукупність всіх ходів називають партією.

Під час гри у кожного гравця можуть виникати певні варіанти дій, послідовність цих дій називаються стратегією. Наприклад у іграх, в яких гравець протистоїть іншому гравцю один на один, послідовність ходів може збігатися, якщо ж виникає така ситуація то і стратегії цих гравців також збігаються. Найкращою стратегією для гравця, буде виступати така, коли при повторенні однакових ходів в багатьох партіях, гравець має максимальний середній прибуток або мінімальний середній програш.

Також слід пам’ятати, що стратегія, що в своїй суті має лише отримання максимального виграшу та не звертає увагу на інші критерії, не володіє оптимальністю до рівноваги. Результатом грим в якій жодна сторона не хоче змінювати кінцевий результат називають стійким рішенням.

Закріпимо, що головною ідеєю теорії ігор є знаходження оптимальної стратегії.

1) зважаючи на тип ходу гру можна поділити на стратегічну та азартну. До азартних ігор належать тільки ті в яких кожен хід має певну випадковість, такі ходи теорія ігор не може прорахувати. Якщо ж кожен хід був прорахований наперед або всі ходи належать гравцю, то таку гру називають стратегічною.

2) ігри поділяються на два типи, ігри один на один, та ігри групами.

3) в залежності від поведінки гравців.В залежності від об’єднання гравців в групи по декілька чоловік та взаємодії в цих групах ігри можуть бути безкоаліційні, коаліційні та кооперативні. Безкоаліційними партіями називають ігри в яких

гравцям заборонено створювати групи, складати угоди, та ціллю кожного кравця є отримання персонального виграшу, що буде більшим ніж у інших учасників гри.

Гра в якій гравці не переслідують отримання персонального максимального виграшу, а працюють на групу без подальшої ідею поділити вигране називають коаліційними.

Так як в кооперативних іграх головна увага не спрямована на перевірку котра стратегія була кращою, що можна прослідити в без коаліційній гри, а більше на отримання прибутку та поділ його в майбутньому між учасниками групи та носить більш складний характер.

4) за наявність різної кількості стратегій, поділяються на скінченну та безкінечну гру. Скінченною грою називають таку кількість можливих ходів коли вони закінчаться, а нескінченною коли не закінчаться.

5) за можливістю перегляду історії ходів. Це означає, що кожен гравець може повністю дізнатися інформацію про всі попередні ходи супротивника, такий випадок називають грою з повною інформацією. Так якщо гравці не мають такої можливості такі ігри називають – іграми з неповною інформацією.

6) за описом гри. Якщо гра представлена у вигляді дерева вона називається позиційною. Якщо ж гра в якій кожен користувач робить ходи послідовно то таку гру назвуть грою в нормальній формі.

7) якщо під час гри сума виграшу буде дорівнювати нулю, при нескінченній кількості гравців, то кажуть про гру з нульовою сумою. Також слід розуміти, що в партії в якій один з гравців має повну домінацію та повний виграш, що дорівнює усім ресурсам супротивника називають антагоністичною.

1.3 Вирішення ігор в чистих та змішаних стратегіях [1]

Змішані стратегії – це матрична гра яка містить сідлову точку, та її рішення знаходиться за принципом мінімаксу. Якщо ж платіжна матриця не має сідлових точок, то застосування мінімаксних стратегій кожним з гравців показує, що гравець

I забезпечить собі виграш не менше a, а гравець II забезпечить собі програш не більш b. Так як a <b, то гравець I прагне збільшити виграш, а гравець II зменшити програш.

Якщо інформація про дії протилежної сторони буде відсутня, то гравці будуть багаторазово застосовувати чисті стратегії випадковим чином з певною ймовірністю. Така стратегія в теорії ігор називається змішаною стратегією. Якщо гравець постійно повторює одні і ті самі ходи, та має під час гри одні і ті самі умови то така стратегія називається змішаною.. Для застосування змішаних стратегій повинні бути наступні умови:

1) в грі відсутня сідлова точка;

2) гравцями використовується випадкова суміш чистих стратегій з відповідними можливостями;

3) гра має однаковий результат;

4) кожного ходу, гравець тримає свою стратегію в таємниці;

5) можливий усереднений результат ігор.

Основна теорема теорії ігор Дж. Фон Неймана: Будь-яка парна кінцева гра з нульовою сумою має, принаймні, одне рішення, можливе серед змішаних стратегій.

Звідси випливає, що кожна кінцева гра має ціну, яку позначимо через g, середній виграш, який припадає на одну партію, що задовольняє умові a < g < b. Кожен гравець при багаторазовому повторенні гри, дотримуючись змішаних стратегій, отримує більш вигідний для себе результат. Оптимальне рішення гри в змішаних стратегіях має наступну властивість: кожен з гравців не зацікавлений у відході від своєї оптимальної змішаної стратегії, якщо його противник застосовує оптимальну змішану стратегію, так як це йому невигідно.

Чисті стратегії гравців в їх оптимальних змішаних стратегіях називаються Активними.

Теорема про активні стратегії. Застосування оптимальної змішаної стратегії забезпечує гравцеві максимальний середній виграш (або мінімальний середній програш), рівний ціні гри G, незалежно від того, які дії робить інший гравець, якщо тільки він не виходить за межі своїх активних стратегій.

Змішані стратегії гравців S1 і S2 позначимо відповідно A1, A2, ... Am і B1, B2, B3 ... Bn, а ймовірності їх використання через pA = (p1, p2, ..., pm) і qB = (q1, q2,. .., qn), де pi ³ 0, qj ³ 0, при цьому:

$$\sum_{i=1}^{m} p_i = 1, \quad \sum_{j=1}^{n} q_i = 1 \qquad (1)$$

Тоді змішана стратегія гравця I - SI, що складається з стратегій A1, A2, ..., Am, має вигляд:

$$S_1 = \begin{pmatrix} A_1, A_2, \ldots, A_m \\ p_1, p_2, \ldots, p_m \end{pmatrix} \qquad (2)$$

Відповідно для гравця II:

$$S_2 = \begin{pmatrix} B_1, B_2, \ldots, B_m \\ q_1, q_2, \ldots, q_m \end{pmatrix} \qquad (3)$$

Знаючи матрицю A для гравця I можна визначити середній виграш (математичне очікування):

$$M(A, \bar{p}, \bar{q}) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} p_i q_j \qquad (4)$$

Гравець I, застосовуючи свої змішані стратегії, прагне збільшити свій середній виграш, досягаючи

$$\beta = \min_{q} \max_{p} M(A, \bar{p}, \bar{q}) \qquad (5)$$

Гравець II домагається:

$$\alpha = \max_{p} \min_{q} M(A, \bar{p}, \bar{q})$$

(6)

Позначимо через p_A^* і q_B^* вектори, відповідні оптимальним змішаним стратегіям гравців I і II, при яких виконується рівність:

$$\min_q \max_p M(A, \bar{p}, \bar{q}) = \max_p \min_q M(A, \bar{p}, \bar{q}) = M(A, p_A^*, q_B^*) \tag{7}$$

При цьому виконується умова:

$$M(A, \bar{p}, q_A^*) \le M(A, p_A^*, q_B^*) \le M(A, p_A^*, \bar{q}) \tag{8}$$

Вирішити гру – це означає знайти ціну гри та оптимальні стратегії. Розглянемо найбільш простий випадок кінцевої гри 2 '2 без сідлової точки з матрицями:

$$S_1 = \begin{pmatrix} A_1, A_2 \\ p_1, p_2 \end{pmatrix}, \quad S_2 = \begin{pmatrix} B_1, B_2 \\ q_1, q_2 \end{pmatrix} \tag{9}$$

З платіжною матрицею

$$A = \|a_{ij}\| = \begin{pmatrix} a_{11}, a_{12} \\ a_{21}, a_{22} \end{pmatrix} \tag{10}$$

Потрібно знайти оптимальні змішані стратегії гравців $S_2^* = (q_1^*, q_2^*) \quad (q_1 + q_2 = 1)$ і ціну гри g.

Хоч би якими були дії противника, виграш буде дорівнює ціні гри g. Це означає, що якщо гравець I дотримується своєї оптимальної стратегії $S_1^*(p_1, p_2)$, то гравцеві II немає сенсу відступати від своєї оптимальної стратегії $S_2^*(q_1, q_2)$.

У грі 2 '2, яка не має сідлової точки, обидві стратегії є активними. Для гравця I маємо систему рівнянь:

$$\begin{cases} a_{11}p_1 + a_{21}p_2 = \gamma \\ a_{12}p_1 + a_{22}p_2 = \gamma \\ p_1 + p_2 = 1 \end{cases} \tag{11}$$

Для гравця II аналогічно:

$$\begin{cases} a_{11}q_1 + a_{21}q_2 = \gamma \\ a_{12}q_1 + a_{22}q_2 = \gamma \\ q_1 + aq_2 = 1 \end{cases}$$

(12)

Якщо g ¹ учасників гри мають тільки повторні ходи, то визначник матриці не дорівнює нулю, отже, ці системи мають єдине рішення.

Вирішуючи систему рівнянь (11) і (12) знаходимо оптимальні рішення, і g:

$$\left.\begin{aligned} p_1^* &= \frac{a_{22}-a_{21}}{a_{11}+a_{22}-a_{21}-a_{12}} \\ p_2^* &= \frac{a_{11}-a_{12}}{a_{11}+a_{22}-a_{21}-a_{12}} \\ q_1^* &= \frac{a_{22}-a_{12}}{a_{11}+a_{22}-a_{21}-a_{12}} \\ q_2^* &= \frac{a_{11}-a_{21}}{a_{11}+a_{22}-a_{12}-a_{21}} \end{aligned}\right\} \quad (13)$$

$$\gamma = \frac{a_{11}a_{22}-a_{12}a_{21}}{a_{11}+a_{22}-a_{12}-a_{21}} \quad (14)$$

Приклад: Дана платіжна матриця:

$$A = \begin{pmatrix} 6 & -2 \\ 3 & 5 \end{pmatrix} \quad (15)$$

Знайти рішення.

Рішення. Так як a = 3, b = 5, то a ¹ b, то і матриця гра не має сідлової точки. Отже, рішення шукаємо в змішаних стратегіях. Запишемо системи рівнянь:

Для гравця II:

$$\begin{cases} 6p_1 - 2p_2 = \gamma \\ 3p_1 - 5p_2 = \gamma \\ p_1 - p_2 = 1 \end{cases} \quad (16)$$

Для гравця II:

$$\begin{cases} 6q_1 - 3q_2 = \gamma \\ -2q_1 - 5q_2 = \gamma \\ q_1 - q_2 = 1 \end{cases} \quad (17)$$

Вирішивши ці системи знаходимо:

$$p_1^* = \frac{7}{10}, \quad p_2^* = \frac{3}{10}, \quad q_1^* = \frac{1}{5}, \quad q_2^* = \frac{4}{5}, \quad \gamma = \frac{18}{5}. \quad (18)$$

Отже оптимальні стратегії гравців мають вигляд:

$$S_1^* = \left(\frac{7}{10}, \frac{3}{10}\right), \quad S_2^* = \left(\frac{1}{5}, \frac{4}{5}\right) \tag{19}$$

Чисті стратегії коли метою учасників матричної гри є вибір найбільш вигідних стратегій, що доставляють гравцеві А максимальний виграш, а гравцеві В мінімальний програш.

Визначення. Чисту стратегію Аі гравця А називають оптимальною, якщо при її застосуванні виграш гравця А не зменшується, якими б своїми стратегіями не користувався гравець В. Оптимальною для гравця В називають чисту стратегію Bj, при використанні якої програш гравця В не збільшується, які б стратегії не застосовував гравець А.

При пошуку оптимальних стратегій спираються на основний принцип теорії ігор – принцип обережності, відповідно до якого кожен гравець, вважаючи партнера по грі досить розумним суперником, вибирає свої стратегії з урахуванням того, що противник ні в якому разі не втратить жодної можливості використовувати будь-яку його помилку в своїх інтересах. Тому гравці повинні бути гранично уважні при виборі кожен своєї чистої стратегії.

Припустимо, що гравцеві А слід зробити свій вибір. Аналізуючи платіжну матрицю (aij) i = 1, m j = 1, n він для кожної чистої стратегії Аі спочатку знайде мінімальне значення очікуваного виграшу:

$$\alpha_i = \min_j \{a_{ij}\} \tag{20}$$

А потім з усіх α_i виділить найбільшу:

$$\alpha = \max_i \{\alpha_i\} \tag{21}$$

І вибере відповідну йому чисту стратегію Аі. Це і буде найбільш краща в даних умовах стратегія гравця А. Її називають максимінною, оскільки вона відповідає величині

$$\alpha = \max_i \min_j \{a_{ij}\} \tag{22}$$

Визначення. Число, яке визначається за цією формулою називається нижньою чистою ціною гри (Максиміна). Воно показує, який мінімальний виграш може

отримати гравець А, правильно застосовуючи свої чисті стратегії при будь-яких діях гравця В.

Основною ціль гравця В – це спираючись на свою стратегію та оптимальність ходів як умога сильніше вплинути на кінцевий результат гравця А. А. Тому для гравця В відшукується число

$$\beta_j = \max_i \{a_{ij}\} \quad (23)$$

Тобто, визначається максимальний виграш гравця А, за умови, що гравець В застосує свою J чисту стратегію, потім гравець В відшукує таку свою стратегію, при якій гравець А отримає мінімальний виграш, тобто знаходить число

$$\beta = \min_j \max_i \{a_{ij}\} \quad (24)$$

Визначення. Число, яке визначається за формулою (24), називається чистою верхньою ціною гри (МІНІМАКСІ) і показує, який мінімальний програш за рахунок своїх стратегій може собі гарантувати гравець В (що відповідно визначає максимальний виграш гравця А при виборі правильних стратегій гравцем В).

Іншими словами, застосовуючи свої чисті стратегії гравець А може забезпечити собі виграш не менше $\underline{\alpha}$, а гравець В використовуючи свої ідеальні ходи, повинен не дати можливості гравцю А отримати більший прибуток ніж $\overline{\alpha}$.

Сідловою точкою чистих стратегій називають випадок якщо a=b, а чиста ціна гри v = $\alpha = \beta$.

Сідлова точка – визначає пару чистих стратегій (io, jo), що досягають рівня $\alpha = \beta$ для гравців А і В. Це значить, що для отримання найкращого результату якогось гравця, він повинен дотримуватись своєї послідовності ходів і це призводить до того, щоб іншому гравцю отримати такий же самий результат, йому потрібно також дотримуватись такої самої послідовності як у першого гравця, а саме:

$$a_{ijo} \le a_{iojo} \le a_{iojo} \quad (25)$$

Де I, j – будь-які чисті стратегії відповідно гравців 1 і 2; (Io, jo) - стратегії, що утворюють сідлову точку.

Виходячи з цього сідловий елемент $a_{i_0 j_0}$ є мінімальним в іо-му рядку і максимальним в jо-м стовпці в матриці А. Відшукання сідлової точки матриці А відбувається наступним чином:

Приклад 1 [3]

$$A = \begin{pmatrix} 1 & -3 & -2 \\ 0 & 5 & 4 \\ 2 & 3 & 2 \end{pmatrix} \begin{matrix} \min\limits_j a_{ij} \\ \| \\ -3 \\ 0 \\ 2 \end{matrix} \Big\} \max_i \min_j a_{ij} = 2$$

$$\max_i a_{ij} = \underbrace{2 \quad 5 \quad 4}$$

$$\min_j \max_i a_{ij} = 2$$

Сідловою є пара (Іо = 3; jо = 1), при якій ціна гри $v = \alpha = \beta = 2$.

Зауважимо, що хоча виграш також дорівнює 2, вона не є сідловою, так як даний результат не є максимальним сере представлених стовпчиків.

Приклад 2 [3]

Аналізуючи платіжну матрицю гри, можна знайти нижню чисту ціну гри – мінус 2. Таким чином, для гравця А максиминной стратегією буде А2 – запис чисел 1 і 2. Верхня чиста ціна гри дорівнює двом, а мінімаксної стратегією для гравця В є В2 – запис такої ж числовий комбінації. Так що з розумних міркувань ця гра приречена на нічию.

Приклад 3[3]

$$H = \begin{pmatrix} 10 & 30 \\ 40 & 20 \end{pmatrix} \begin{matrix} \min\limits_j a_{ij} \\ \to \\ \to \end{matrix} \begin{matrix} 10 \\ 20 \end{matrix} \Big\} \max_i \min_j a_{ij} = 20$$

$$\max_i a_{ij} \quad \begin{matrix} \downarrow & \downarrow \end{matrix}$$

$$\underbrace{40 \quad 30}$$

$$\min_j \max_i a_{ij} = 30$$

З аналізу платіжної матриці Н видно, що $\alpha < \beta$, тобто дана матриця не має сідлової точки. Якщо гравець А вибирає свою чисту максиминну стратегію I = 2, то гравець В, вибравши свою мінімаксну J = 2, програє тільки 20. В цьому випадку гравцеві А вигідно вибрати стратегію і = 1, тобто відхилитися від своєї чистої максиминной стратегії і виграти 30. Тоді гравцеві В буде вигідно вибрати стратегію

j = 1, тобто відхилитися від своєї чистої мінімаксної стратегії і програти 10. У свою чергу гравець А повинен вибрати свою 2-ю стратегію, щоб виграти 40, а гравець В відповість вибором 2-й стратегії і т.д.

Теорема 1. В матричній грі нижня чиста ціна гри не перевищує верхньої чистої ціни гри, тобто $\alpha \le \beta$.

Доказ. За визначенням $\alpha_i = \min_j |a_{ij}| \le a_{ij}$; $\beta_j = \max_i |a_{ij}| \ge a_{ij}$. Об'єднуючи ці співвідношення, отримаємо $\alpha_i = \min_j |a_{ij}| \le a_{ij} \le \max_i |a_{ij}| \le \beta_j$. Звідси $\alpha_i \le a_{ij} \le \beta_j$ або $\alpha_i \le \beta_j$. Ця нерівність справедлива при будь-яких комбінаціях і і j. Буде вона справедливою і для тих і і j, для яких реалізуються максимальне і мінімальне значення і. Що і потрібно було довести.

Щоб дослідити матричні ігри треба почати з знаходження сідловок точки матриці в ідеальних умовах. Якщо така точка була знайдена в ідеальних умовах, то дослідження даної матриці, а також і гри буде закінчено. У випадку коли точку знайти неможливо, звичайно в ідеальних умовах, треба дивитися на максимальний та мінімальний прибуток гри. Дані границі будуть вказувати, що гравець не може розраховувати на отримання більшого прибутку, ніж максимальне значення гри, але розраховувати на більше ніж мінімальне значення. Для покращення рішення гри слід звернути увагу на використання стратегій та на можливість повторення їх в партії. Результат який буде отримано, буде досягнуто у випадку коли умови були ідеальними.

1.4 Графічне вирішення ігор лінійного програмування

Вирішення ігор лінійного програмування графічним методом розглянемо на прикладі.

Приклад 1

Вирішити графічно це завдання лінійного програмування.

$$F = 4x_1 + 3x_2 - 1 \to \max$$

$$\begin{cases} x_1 + x_2 \le 8 \\ -2x_1 + 3x_2 \le 9 \\ 2x_1 - x_2 \le 10 \end{cases}$$

$$x_1 \ge 0,\ x_2 \ge 0$$

Рішення. Знайдемо спочатку область допустимих рішень (ОДР). Вирішимо графічно першу нерівність:

$$x_1 + x_2 \le 8 \tag{26}$$

Для цього побудуємо спочатку пряму лінію, відповідну рівняння:

$$x_1 + x_2 = 8 \tag{27}$$

Якщо $x_1 = 0,$ то $x_2 = 8,$ то пряма (26) проходить через точку М1(0;8). Аналогічно, якщо $x_1 = 8,$ то $x_2 = 0,$ і пряма (27) проходять также через точку М2 (8;0). Проведемо через ці дві точки прямую лінію і відмітимо її за допомогою 1 (див. рис. 1). Ця лінія ділить площину на дві півплощини, які ми умовно назвемо верхня і нижня півплощини. Так як координати точки (0; 0) задовольняють нерівності (1), то цій нерівності відповідає нижня напівплощина, яка містить цю точку. Цей факт ми покажемо на рис. 1 штрихами, спрямованими вниз від лінії 1.

Тепер вирішимо графічно другу нерівність:

$$-2x_1 + 3x_2 \le 9. \tag{28}$$

Йому відповідає пряма, задана рівнянням:

$$-2x_1 + 3x_2 = 9 \tag{29}$$

Її ми побудуємо трохи інакше. Перепишемо рівняння (29) у вигляді:

$$x_2 = \frac{2}{3}x_1 + 3$$

Тоді при $x_1 = 0$ виявляється, що $x_2 = 3$, що дає точку М3 (0;3) шуканої прямої. Кутовий коефіцієнт цієї прямої $\kappa = \frac{2}{3}.$. Але кутовий коефіцієнт будь-якої прямої дорівнює $tg\alpha,$ де α- кут нахилу прямої до осі 0х: $tg\alpha = \kappa = \frac{2}{3}$.

Якщо тепер ми відкладемо три одиниці вправо від точки М3 (0; 3) і потім дві одиниці вгору, то отримаємо іншу точку М4 (3; 5) яка також лежить на прямій (29).

Через точки М3 і М4 ми проводимо пряму 2 (рис.1). Початок координат (0; 0) задовольняє (28) і лежить нижче графіка лінії 2, тому відповідна напівплощина є «нижньою, що ми і відзначаємо штрихами, спрямованими вниз від прямої 2 (рис.1). Аналогічно будуємо пряму 3.

Рівняння

$$2x_1 - x_2 = 10 \tag{30}$$

Замінюємо на рівняння $x_2 = 2x_1 - 10$.

Ясно, що пряма проходить через т. М5 (5; 0), і має кутовий коефіцієнт до = 2. При цьому самій нерівності

$$2x_1 - x_2 \le 10 \Leftrightarrow x_2 \ge 2x_1 - 10$$

Відповідає верхня напівплощина, відмічена штрихами вверх від прямої 3.

Тривіальній нерівності $x_1 \ge 0$ відповідає права напівплощина координатної площини, тобто напівплощина, що лежить праворуч від вертикальної осі $0x_2$. Її відзначаємо штрихами, спрямованими вправо від осі 0. Нарешті нерівності відповідає верхня напівплощина координатної площини, зазначена штрихами, спрямованими вгору від осі 0. Перетин всіх зазначених напівплощин визначає ОДР даного завдання. На малюнку 1 це область, обмежена опуклим п’ятикутником OABCD.

Зобразимо на рисунку 1 вектор $\bar{c}$ зростання цільової функції *F*. Це вектор $\bar{c}$ початком в т. (0; 0) і кінцем в точці М (4; 3), оскільки $\bar{c}$ = 4,3.

Побудуємо тепер лінію рівня $F(x) = 11$. Вона визначається рівнянням:

$$4x_1 + 3x_2 = 12. \tag{31}$$

Ми взяли тут константу С = 11, для того щоб точки перетину прямої (4) з осями x_1 та x_2 мали цілі координати. Дійсно, якщо $x_1 = 0,$ то $x_2 = 4,$ і якщо $x_2 = 0,$ то $x_1 = 3,$. Це дає дві точки М1 (0; 4) і М2 (3; 0) лінії рівня (31). Через них проводимо пунктиром відповідну лінію рівня (рис. 1). Вона виявляється перпендикулярна вектору зростання. Відрізок перетинається з ОДР і в кожній його точці х значення цільової функції дорівнює 11:

$$F(\bar{x}) = 11, \quad x \in [M_1; M_2] \tag{32}$$

Ми знаємо, що значення функції F збільшується в напрямку вектора зростання $\bar{c}$. Щоб знайти максимальне значення $F(x)$ на ОДР будемо паралельно переміщати лінію рівня в напрямку вектора зростання $\bar{c}$. До тих пір, поки, вона буде мати хоча б одну точку перетину з ОДР 1 ясно, що останнім перетином зміщеною лінії рівня (31) буде точка С (рис. 1.1).

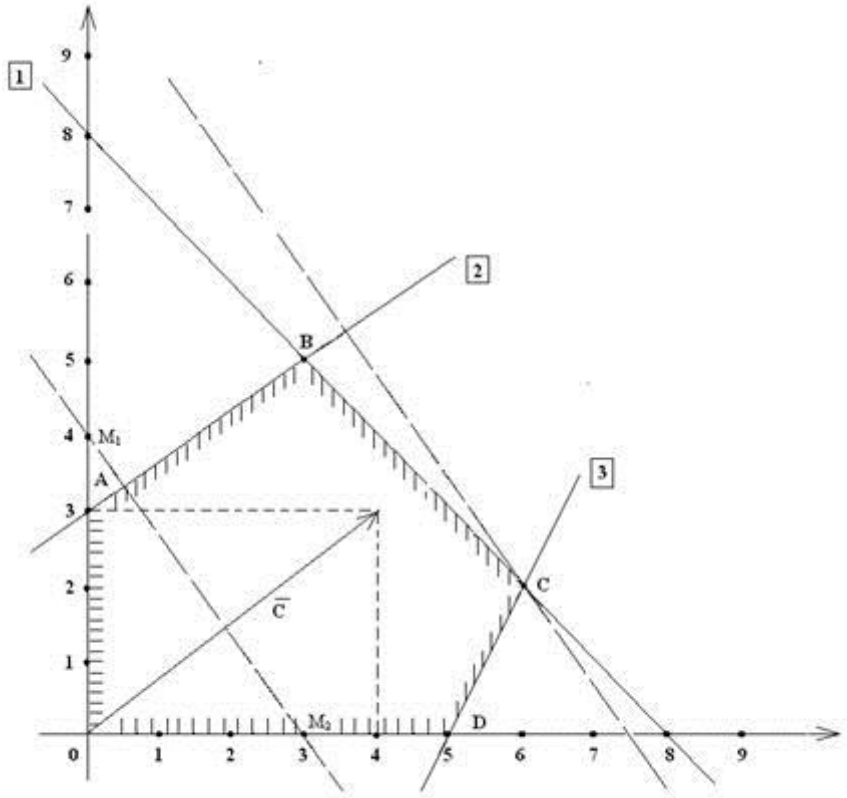

Рисунок. 1.1. Графічне рішення завдання ЛП

На цій лінії очевидно і буде досягатися максимальне значення цільової функції F в ОДР, оскільки при подальшому русі лінії рівня в напрямку вектора зростання, вона перестає перетинатися з ОДР. Отже, максимальне значення функція F (x) має в точці *C*. Так як точка *C* є перетином прямих 1 і 3, то її координати знаходяться з системи:

$$\begin{cases} x_1 + x_2 = 8 \\ 2x_1 - x_2 = 10 \end{cases} \tag{32}$$

Щоб вирішити цю систему, складемо обидва рівняння. Тоді отримаємо, що $3x_1 = 18$ або $x_1 = 6$.

З першого рівняння знаходимо, що $x_2 = 8 - x_1 = 8 - 6 = 2$.

Отже, координати точки С знайдені: С (6; 2). Знайдемо максимальне значення функції:

$$F_{max} = F(C) = F(6;2) = 4 \times 6 + 3 \times 2 - 1 = 29.$$

Задача вирішена.

Відповідь: максимальне значення цільової функції F досягається в точці С (6; 2) і так само 29:

$$F_{max} = F(6;2) = 29 \quad (33).$$

1.5 Висновок

В даному розділі були розглянуті основні поняття в теорії ігор та поняття гри. Також були визначена мета теорії ігор, а саме знаходження розуміння поведінки учасників конфлікту. Були виведені деякі обмеження, а саме припущення про повне розуміння супротивника під час конфлікту, ще було встановлені деякі недоліки теорії ігор, а саме те, що кожному гравцю потрібно знати всі можливі дії супротивника.

Також були розглянуті деякі стратегії та умови їх використання, а саме змішаної стратегії та чистої стратегії, ще були наведені приклади вирішення за стратегіями.

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

2.1 Поняття CRM системи [4]

CRM – це система, що спрямована на створення довготривалих і привабливих взаємин з замовником через персональний аналіз потреб. Окрім цього, це вже не нова стратегія, що спрямована на покращення і підвищення прибутковості бізнесу компанії, шляхом підвищення лояльності клієнта протягом усього циклу взаємодії з ними. Цикли включає глибокий синтез стратегічного передбачення, корпоративного розуміння етики споживача в умовах багатоканальних дистриб'юторських ринків, приведення коштів, управління даними розроблених під CRM додатки, а також якісних операцій, старанності і сервісу. В CRM підкреслюється, що управління взаємовідносинами з клієнтами – багатоплановий і тривалий процес, відображення і реакція на швидко мінливе бізнес-середовище.

До виникнення CRM як різновиду менеджменту призвело кілька важливих тенденцій, серед них:

- зрушення орієнтації бізнесу від транзакційного до маркетингу взаємовідносин;
- поступове розуміння, що клієнти є активами бізнесу, а не просто рекламної аудиторією;
- перехід в стратегічному структуруванні компанії від функцій до процесів;
- визнання вигоди від використання інформації для попередження подій, а не тільки для проходження за ними;
- більш широке використання технологій для управління інформацією і максимізації її цінності;
- визнання необхідності компромісу між обслуговуванням клієнтів і витягом;
- розвиток індивідуальних маркетингових підходів.

Метою CRM є залучення та утримання вигідних клієнтів за допомогою встановлення і поліпшення відносин з ними. Розробка стратегії взаємовідносин з клієнтами стала можливою завдяки проривів в області інформаційних технологій (IT). Такий розвиток подій дозволяє створювати великі масиви клієнтських даних,

гарантує широку зворотний зв’язок з клієнтами і дозволяє аналізувати та використовувати дані. Також слід зазначити, що вартість обслуговування знижується за рахунок більшої кількості переваг. Цей надлишок доступних інструментів для CRM дозволяє компаніям набагато ефективніше націлюватися на самі напрямки.

Нижче розглянуто основні типи CRM.

Операційний CRM [5] – це вид систем, що націлений на повну автоматизацію та покращення процесу роботи з клієнтом. Включає в себе автоматизацію комунікації, розміщення та реклами товару . Нажаль, під впливом автоматизації основною частиною витрат стали саме дані системи, оскільки бездумно встановлювалися в компаніях. Постачальники систем CRM привертають увагу компаній, пропонуючи все більш і більш широкий спектр операційних розробок CRM.

Аналітичний CRM включає в себе пошук, накопичення, організацію, аналіз, інтерпретацію і використання даних, отриманих в операційній частині бізнесу. Дуже важливо розглянути можливість інтеграції методів аналітичного CRM з методами операційного CRM.

Спільний CRM включає в себе використання спільних сервісів та інфраструктури, щоб зробити можливим взаємодію компанії з її численними каналами. Даний тип систем полегшує комунікацію між представниками компанії та замовником.

Об’єднавши три компоненти, ми отримаємо модуль, я кому кожна частина живить одна одну; їх інтеграція просто необхідна для успішної роботи CRM, яке виражається в поліпшенні досвіду взаємовідносин з клієнтами. Спільний CRM дозволяє клієнтам контактувати з компанією використовуючи велику кількість різних модулів та швидко швидко отримувати відповіді на поставленні запитання. Системи CRM спрощують зв’язок замовників та компанії, а також подальше виконання вимог замовника. Аналітичний CRM допомагає націлюватися на потрібних клієнтів і пропонувати їм деякі можливості, а також за допомогою набагато більш високого рівня знань створювати для кожного з клієнтів унікальну

модель маркетингу. В наш час все більше компанії дивляться в бік аналітичних систем, оскільки потребують оптимізації в роботі з клієнтами та побудові по шагової вигоди для себе так і для замовника.

На ринку CRM використовуються також такі терміни:

Стратегічний CRM, що.включає в себе розвиток підходу до CRM, що базується на бізнес-стратегії підприємства і націленого на розвиток взаємин з клієнтами, які дають результат у вигляді довгострокового зростання прибутку акціонерів.

e-CRM – це термін, що відноситься до використання в CRM засобів електронної торгівлі або електронних каналів збору даних.

Маркетинг взаємовідносин з партнерами – це термін, що означає CRM-діяльність, спрямовану на стратегічних партнерів або репеллерів. Більшість IT-компаній працюють через непрямі канали, а значить, МВП з посередниками є ключовим елементом CRM-програми продавця.

Види CRM-стратегії [5]

Бізнес - стратегія і клієнтська стратегія – два головні компоненти CRM - стратегії. Центральним в процесі розробки стратегії CRM стане вибір клієнтів і їх характеристик, ступеня сегментації, визначення, якою мірою індивідуальний підхід до сегментації відповідає цілям і засобам компанії, а також оцінка повноти готівкової та потенційно доступної інформації про клієнтів. Такі рішення повинні ґрунтуватися на поточну ситуацію компанії і на ситуації, можливої в майбутньому.

Продукт-орієнтовані продажі.

У цьому випадку компанія має дані про транзакції та бажає провести простий аналіз таких змінних, як кількість продажів продукту в одиницю часу і продуктивність каналів дистрибуції. Хоча може використовуватися і точкова розсилка реклами клієнтам, у компанії мало або зовсім немає детальної інформації про індивідуальних клієнтів.

В рамках продукт-орієнтованих продажів для посилення ефективності маркетингу на основі використання порівняно простих баз даних може бути проведений ряд простих дій. Наприклад, за допомогою операційних систем компанії

можна вдатися до простих засобів аналізу або складання запиту, а для більш тонкого аналізу операційні дані доповнюються (часто з зовнішніх джерел) за допомогою спеціальних засобів. Наприклад, можуть бути проведені наступні аналізи:

- простий аналіз списків рекламної розсилки;

- найпростіша сегментація на основі продукції або каналів дистрибуції;

- прості запити і звіти про продуктивність продажів і продуктивності каналів продажів.

Основна увага на продукт – орієнтовані продажі приділяється не клієнтам, а продукції та каналам продажів. Дуже малоймовірно, що сегментація ринку в даному випадку буде заснована на клієнтах, за винятком компаній, що випускають один єдиний продукт або простий ряд подібних продуктів. Рівень інтелектуальної складової в плані CRM тут досить низький. Проте, такий підхід може стати цілком адекватною стратегією для деяких компаній.

Регульований сервіс і підтримка – це підхід, що стосується застосування сервісу клієнтів до продажу. Компанія намагається поліпшити взаємини з клієнтами за допомогою підвищення рівня сервісу і підтримки, наприклад, через Call - центри або телефонний маркетинг. У цьому різновиді CRM всебічна інформація про клієнтів не потрібна, але комунікація відбувається на особистому, індивідуалізованому рівні.

Зазвичай тут мова йде про використання більш тонких засобів для роботи з досить простими даними про клієнтів, якими є, наприклад:

- центри контакту / служби допомоги;

- телефонний маркетинг;

- управління контактами;

- автоматизація роботи продавців.

До клієнт - орієнтований маркетинг вдаються компанії що, сьогодні зрушують увагу з індивідуальних продажів продукту на клієнтів. При цьому компанія намагається виробити більш детальне розуміння своїх клієнтів, а також провести ряд аналізів, що включають:

- аналіз вигідності клієнтів;

- аналіз дій конкурентів;

- управління лояльністю і «плинністю» клієнтів;

- аналіз клієнтських оцінок компанії;

- аналіз лояльності клієнтів;

- аналіз контролю виявлення підробок;

- аналіз управління ризиками;

- аналіз випадків несплати рахунків.

Не всі з цих аналізів мають відношення до кожної компанії. Їх релевантність буде залежати від галузі, позиції компанії на ринку та інших факторів.

В випадку коли компанія хоче змінити вектор роботи з менеджменту товарів до роботи з клієнтами, їй необхідно аналіз кожного замовника для визначення якого з них вигідніше утримувати, що дозволить:

- знайти універсальний підхід до кожного;

- продивитися весь шлях компанії;

- виявлення нових місць продажу для отримання прибутку з від нових клієнтів.

Хоча клієнт - орієнтований маркетинг є куди більш розвиненою формою CRM, компанії, що працюють в цьому руслі, не зможуть запропонувати клієнтам той високий індивідуалізований сервіс та підтримку, який знаходиться в правій частині матриці.

Індивідуалізований CRM [5] – це платформи, що мають в своїй архітектурі платформу, для аналізу даних та роботи з схожими додатками, в них входять:

- додатки для індивідуального маркетингу, наприклад B2B та B2C;

- сучасні програми комп’ютерної телефонії для інтерактивного використання комп’ютера в ході телефонного контакту з клієнтом і забезпечення його індивідуалізованого обслуговування;

- програми для багатоканальної інтеграції;

Це додатки, що дозволять в спільній роботі з клієнтом або його представником переглядати сторінки онлайн.

Зокрема дана стратегія підходить компаніям, що хочуть мати широкий вибір канальних опцій. У їх число можуть продажі, продажі через посередників (дистриб'юторів чи агентів), продажі через Інтернет за допомогою систем електронної торгівлі. Остання з опцій відкриває великі можливості для вибудовування індивідуалізованих маркетингових систем, які видобувають інформацію з онлайнових взаємодій з клієнтами і вибудовують диференційований сервіс, обслуговуючи кожного окремого клієнта, що входить в систему. На самому високому рівні індивідуалізованого CRM компанія здатна миттєво відповідати на запити клієнтів, і як тільки відбувається взаємодія або транзакція, інформація надходить в операційні системи. При цьому CRM набуває динамічну, а не статичну форму.

Підприємства, що використовують персональні CRM, будуть намагатися дати клієнтам повний, індивідуальний і кастомний сервіс. Клієнти можуть обслуговуватися по оналйн-чатах, через поштову скриню або при особистих зустрічах, або ж за допомогою систем електронної торгівлі в Інтернеті (клієнти при цьому використовують для запитів і покупок браузери).

Цей формат CRM, яка не потребує персонального підходу, тобто, особистого контакту з клієнтом, неодмінно повинна бути індивідуалізованою. Щоб CRM був індивідуалізованим, підприємствам необхідна розробка IT-систем, яка «знає» клієнта, – при цьому вона розвиває корпоративну пам'ять про своїх клієнтів.

Таким чином, CRM-система на даному етапі ведення бізнесу стає невід'ємною його частиною. За допомогою CRM - систем автоматизується робота відділу продажів, накопичується, обробляється інформація про клієнтів, що в свою чергу, дозволяє на багато швидше і якісніше обслуговувати клієнтів. CRM - системи служать підвищенню прямих і крос продажів, за допомогою цього збільшується прибуток компанії.

Для огляду існуючих рішень розглянемо деякі із найбільш популярних систем CRM.

2.2 Опис існуючих рішень

«Бітрікс24» (міжнародна назва Bitrix24) [6] – це програмний комплекс (або сервіс), призначений для оптимізації ведення бізнесу та контролю за виробничим процесом. Відразу хотілося б відзначити, що його привабливість полягає в тому, що не доведеться купувати власний сервер, оскільки всі дані будуть знаходитися в хмарному сховищі. Всі дані добре захищені, і доступ до них матимуть лише авторизовані користувачі.

Система «Бітрікс24» багата на функціональні можливості. Її опис містить інформацію про те, що це програмний комплекс. Основними його функціями є:

- створення корпоративної мережі компанії (тільки після попередньої реєстрації учасників робочої групи);

- контроль за роботою організації (включає в себе формування завдань, контроль за їх виконанням і ведення статистичної звітності);

- сховище для файлів. Залежно від тарифу, який обраний адміністратором компанії, різниться обсяг дискового простору;

- конструктор для створення сайтів;

- додаток для здійснення дзвінків по IP-телефонії.

- інші додаткові можливості (чат, календар, соціальна мережа закритого типу і т. д.). На рис. 2.1 зображено приклад функціоналу.

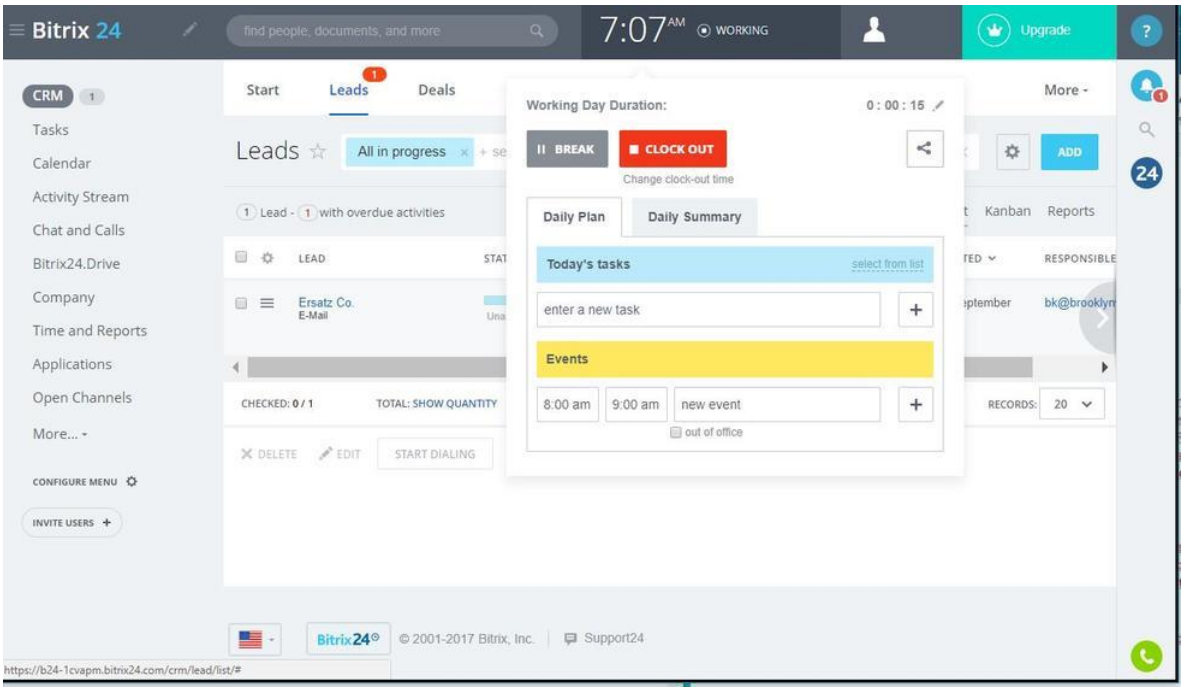

Рисунок 2.1 – Головне вікно Bitrix24

Щоб почати користуватися можливостями Bitrix, необхідно пройти процедуру реєстрації. Тут слід згадати про те, що особливості цього процесу безпосередньо залежать від обраного тарифного плану. На сьогоднішній момент існує 3 тарифи – безкоштовний і 2 платних (Команда і Компанія), сторінка обрання тарифів представленні на рис.2.2. Їх відмінність полягає в тому, що на безкоштовному тарифі може бути зареєстровано не більше 12 осіб. Крім того, він має трохи обмежений функціонал, але для невеликої організації цього більш ніж достатньо. Якщо у компанії працює більше 12 чоловік, то потрібно буде вибирати один з платних пакетів. Процес реєстрації в безкоштовному тарифі досить простий – потрібно лише ввести основні дані про себе і компанії. Ніяких документів надавати не доведеться. Для реєстрації на платному пакеті потрібно буде надати окремі види документації та вказати про себе більш докладні відомості. Власне, на цьому процедура реєстрації в «Бітрікс24» завершена і можна починати користуватися доступним функціоналом.

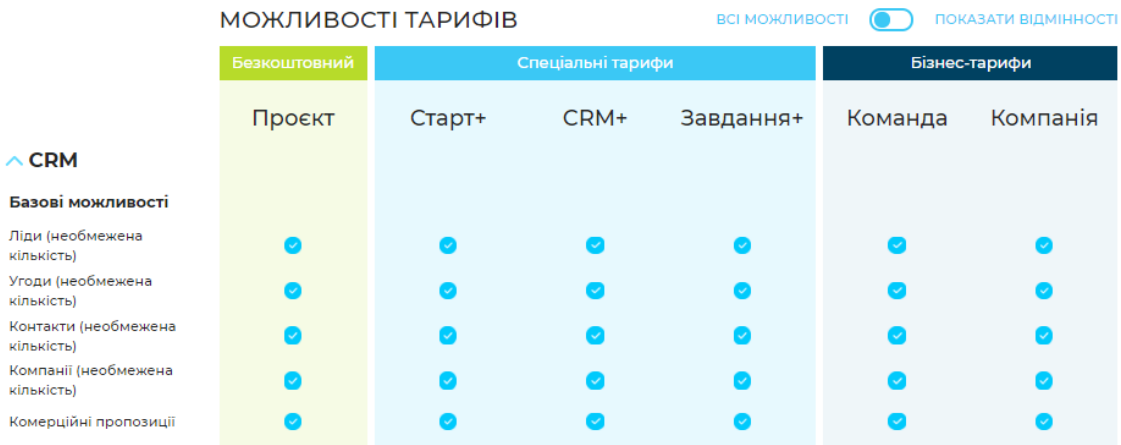

Рисунок 2.2 – Сторінка тарифів Bitrix24

Після завершення реєстрації необхідно привласнити комплекс технічних можливостей для кожного користувача (співробітника компанії). Даний процес фактично є створенням корпоративної соціальної мережі. З усіх зареєстрованих користувачів залишається один основний (так званий системний адміністратор),

який і буде займатися розподілом ролей. Для кожного співробітника можна залишити повний набір можливостей або привласнити окремі ролі. Дане питання вирішується на рівні самої компанії і розробники Bitrix24 не мають до нього ніякого відношення).

Існує можливість переносити всю інформацію з хмарного сховища на свій сервер. Для цього знадобиться окрема програма, яку можна завантажити на сайті розробників «Бітрікс». Однак дане рішення залишається безпосередньо за керівником компанії і залежить багато в чому від особливостей виробничого процесу. Залежно від цілей використання системи кожен може вибирати саме ті функціональні можливості, які йому необхідні. Конструктор сайтів дозволяє створювати унікальний дизайн з подальшим його розміщенням на окремому хостингу і відкриттям до нього загального доступу. Завдяки планувальнику завдань (CRM) і календарю можна контролювати весь робочий процес і формувати статистичну звітність. Правда, в безкоштовному тарифі можливості цього дещо обмежені, але при бажанні наявного функціоналу цілком достатньо для оптимізації та організації контролю на роботі.

Програма «Бітрікс24» дозволяє всім користувачам створювати, розміщувати і редагувати документи безпосередньо в хмарному сховищі. В якості основи для створення розробники використовують функціонал від Google G Suite, MS Office365, MS OfficeOnline, що забезпечує повну сумісність форматів, приклад приведено на рис. 2.3.

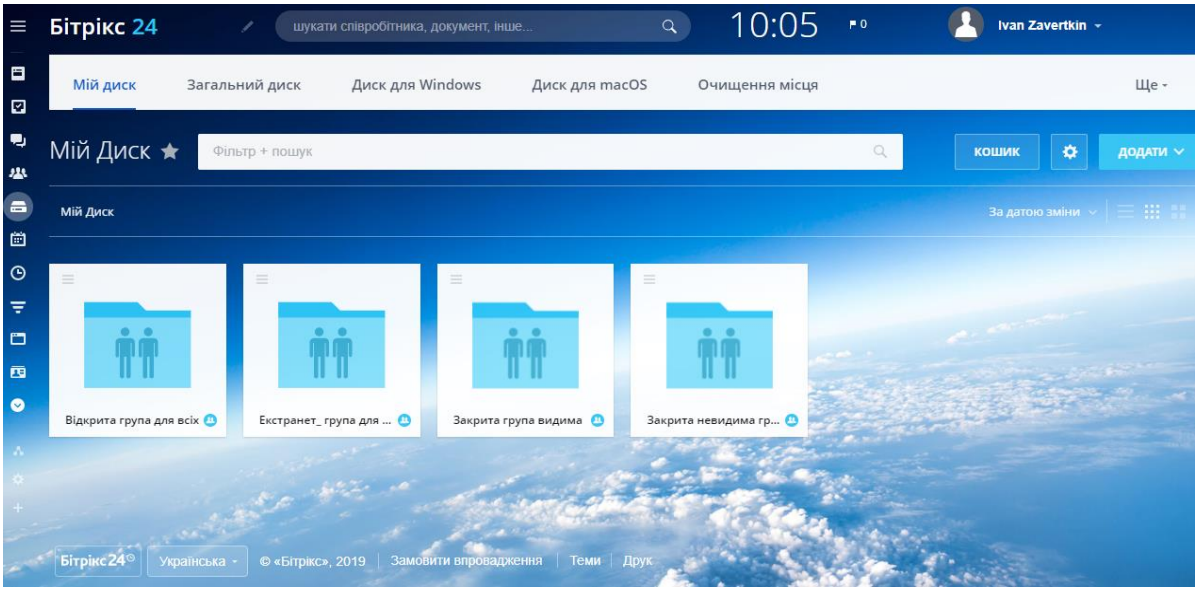

Рисунок 2.3 – Приклад хмарного сховища Bitrix24

Примітно те, що на сьогоднішній момент «Бітрікс24» працює не тільки на російській мові – до нього додано ще і 11 іноземних. Таким чином, дане програмне забезпечення можна сміливо вважати повноцінним міжнародним продуктом. Завдяки цілодобовій роботі підтримки збій в роботі даного програмного комплексу практично виключений. За всю історію розвитку Бітрікс (а це понад 8 років) відбулося всього 2 збої в роботі програми. Користувачі тоді дивувалися, що це відбулося, однак їхні дані в результаті були відновлені в повному обсязі.

Таким чином, «Бітрікс24» є універсальним програмним продуктом для ведення бізнесу. Завдяки єдиному функціоналу підприємці і бізнесмени з усього світу можуть об’єднуватися між собою і вести навіть інтернет-співробітництво. Недоліків у цій програмі практично немає. Завдяки потужним інструментам посібників, IT-фахівці тієї чи іншої компанії можуть обійтися без покупки іншого програмного забезпечення. Весь інтерфейс системи є цілком зрозумілим – з ним може розібратися навіть користувач без досвіду. А завдяки можливостям використання в оффлайн режимі застосовувати програму можна в будь-якій точці світу.

Hubspot CRM [7] – дозволяє автоматизувати взаємодії з клієнтами через електронну пошту або соціальні мережі, відкриваючи масу можливостей для аналітики і зростання продажів, головне вікно на рис.2.4.

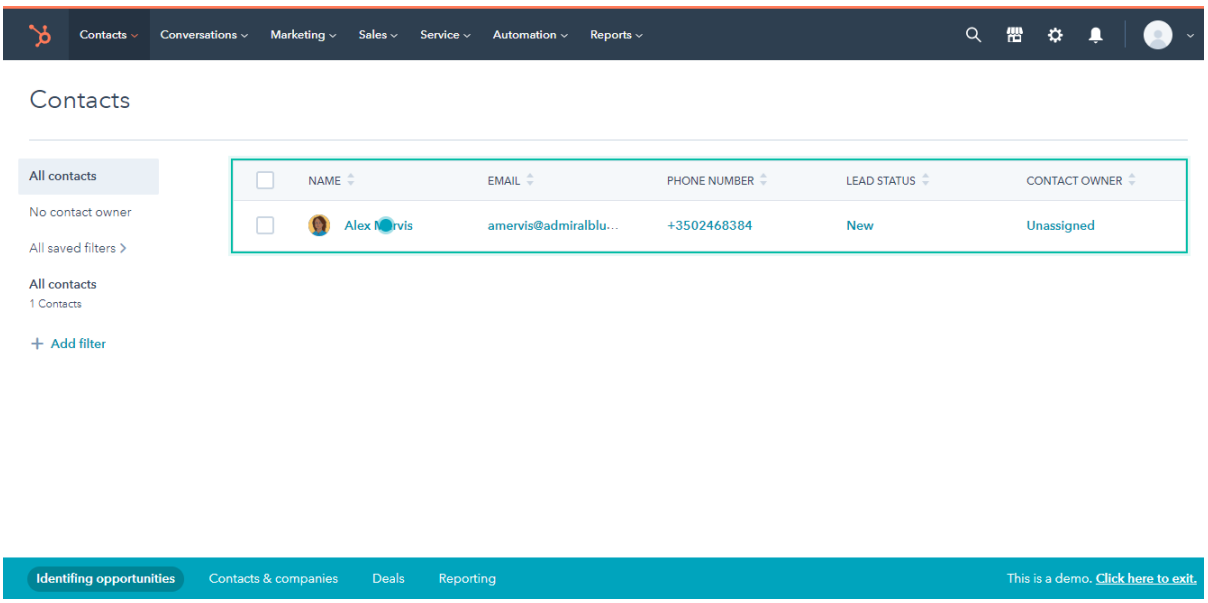

Рисунок 2.4 – Головне вікно Hubspot

Допомога бізнесу від Hubspot CRM зводиться до спрощеного рішенням таких завдань, як:

- ведення клієнтської бази;

- автоматичне відстеження взаємодій з партнерами і покупцями;

- планування і постановка задач;

- аналітика результатів продажів.

Найголовніше, що: це безкоштовно і на необмежений термін; сам сервіс дуже якісний і має інтуїтивно зрозумілий інтерфейс. Хабспот – це рішення для малого бізнесу, яке не потребує вкладень і дозволяє відстежувати ділові зв'язки.

Варто зауважити, що Hubspot повністю англійською мовою і підійде далеко не кожному користувачеві. Тим не менш, вона має необхідні функції, які є у популярних онлайн-CRM.

Ось деякі з них:

Contact Management – управління контактами і базою клієнтів;

Company insights & records – ведення записів компанії;

Gmail & Outlook integration – автоматична синхронізація з поштовими сервісами, що дозволяє фіксувати діалоги з контрагентами;

Team email & scheduling – прив’язка до електронної пошти компанії, а також планування розсилок;

Live chat – установка онлайн-чату на сайт і збереження листувань в ньому в єдиному вікні + можлива настройка бота;

Deals – ведення запису успішних угод;

Tasks – постановка цілей з нагадуваннями для команди;

Ticketing – робота з проблемами клієнтів через систему тікетів (заявок).

Крім того, разом з CRM безкоштовно поставляються кілька унікальних інструментів Sales and Marketing Hub, наприклад, інтеграція з Google Ads та Facebook, сторінку Marketing Hub можна подивитися на рис. 2.5.

Hubspot слідує стандартної політиці конфіденційності, збираючи такі дані користувачів, як: відвідування веб-сайтів; встановлені мобільні додатки; персональну інформацію (e-mail, телефон, ім’я); лог-файли з конфігурацією комп’ютера або іншого пристрою.

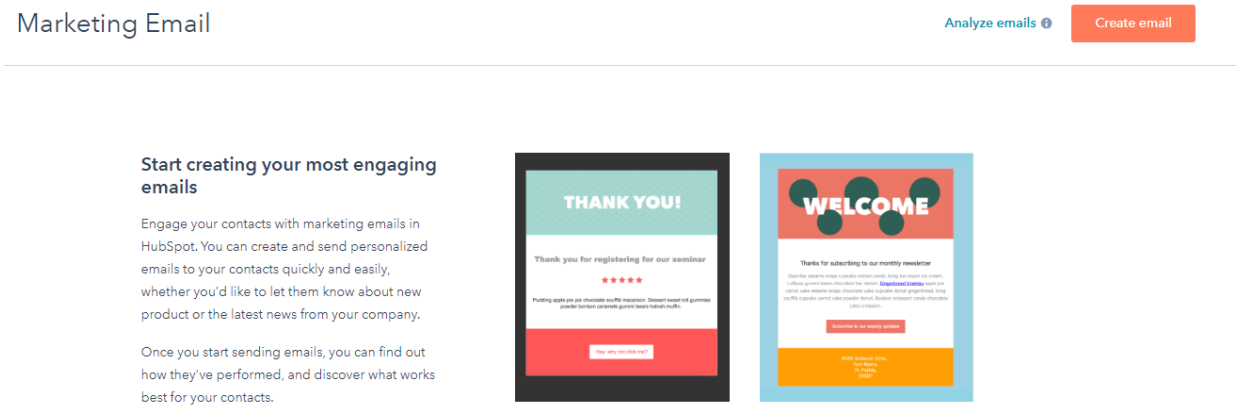

Рисунок 2.5 – Сторінка and Marketing Hub

Згідно з умовами використання на офіційному сайті, Hubspot дотримується двох принципів: суворе дотримання політики конфіденційності; ніколи не передавати персональну інформацію третім особам. Реалізація отриманих даних допускається тільки для персоналізації веб-сайтів у конкретного користувача, а також для відправки релевантних пропозицій по e-mail.

Взаємодія з клієнтами – першочергове завдання CRM. Для цих цілей в Hubspot є два ключових інструменти: Contacts – в цьому розділі зберігаються дані клієнтів (ім'я, email, телефон і т.д.); Conversations – тут здійснюється управління email-листами в єдиному вікні, онлайн-чатом та мессенджерами.

Далі буде розглянуто кожен з перерахованих вище пунктів. Контакти розташовуються в лівому верхньому кутку особистого кабінету. В цілому, інструмент являє типовий набір для ведення клієнтської бази, за аналогією з платними CRM. Доступні кілька настроюваних колонок, рядок пошуку і різні фільтри. Записати в базу безкоштовно можна до 1 000 000 контактів, цього достатньо для середньостатистичного бізнесу.

Інструмент Conversations знаходиться зверху поруч з Contacts. Внизу зліва розташовуються канали зв'язку з клієнтами:

Email – для відстеження листів на електронну адресу компанії.

Chat – налаштування онлайн-чату для сайту і відстеження листувань.

Messenger – підключення бізнес-сторінки Facebook і робота з повідомленнями. У Hubspot є все необхідне для ефективної взаємодії з клієнтами. Можливо фіксувати інформацію по кожному з них в базу даних, і автоматично зберігати листування в чаті, електронною поштою і навіть в Facebook.

Для управління продажами розроблений розділ Sales, розташований в центрі верхнього меню. Він дозволяє виконувати кілька завдань:

Deals – ведення обліку успішних угод;

Tasks – запис поточних цілей з нагадуваннями про важливі події;

Documents – прив’язка до бази контактів різних документів в PDF, Word або Powerpoint;

Meetings – електронний щоденник для компанії, інтегрується з Google або Office 365 calendar.

Перерахованих вище інструментів досить, щоб вести статистику і облік продажів, прикріплювати документацію і нагадування про майбутні взаємодії з контрагентом.

Перейдемо до більш унікальних функцій Hubspot, що виділяють CRM на тлі конкурентів.

Варто особливо відзначити панель Pipeline, де можна в пару кліків додавати успішні угоди. Інструмент знаходиться в розділі Deals, згаданому вище, і дозволяє аналізувати поточну діяльність кожного співробітника, наприклад, його продуктивність, співвідношення проведених вдалих і програних угод і т. д.

На перший погляд може здатися, що в дошці з декількома успішними (або навпаки) продажами немає нічого привабливого, але не поспішайте з висновками. У Hubspot наочно відображається динаміка прогресу компанії на графіках, на основі даних з Pipeline.

В підсумку, користувач отримує просунуту аналітику свого бізнесу і бачить ступінь досягнення фінансової мети.

Автоматична реєстрація активності продажів Hubspot CRM автоматично фіксує взаємодії з контрагентами за доступними каналами зв'язку: телефонні дзвінки; Gmail, Outlook; Facebook; онлайн-чати; звіти про особисті зустрічі.

При здійсненні продажу без CRM взаємодії з клієнтами дуже часто губляться. Наприклад, якщо один співробітник звільниться, то його листування в чатах, email і соціальних мережах можуть просто зникнути, а новому менеджеру доведеться перепитувати і викликати роздратування у контрагентів. Така ситуація веде до низького ступеня задоволеності сервісом і багатьом іншим проблемам.

Провідна інформація в одному місці Інструмент Activity Feed відстежує дії кожного співробітника компанії і публікує їх для загального огляду. Наприклад, якщо одному менеджеру зустрівся проблемний клієнт, то інші колеги побачать це і не стануть витрачати час на нього наступного разу. Крім того, по кожному контрагенту буде видно весь ланцюжок дій і листувань, що дозволить швидко проаналізувати ситуацію при необхідності.

Отже, Hubspot CRM [8] – це безкоштовна англомовна система для управління взаємодіями з клієнтами. Вона володіє стандартними функціями: складання клієнтської бази, менеджмент продажів, ведення записів і підключення необмеженого числа співробітників. Серед унікальних можливостей CRM варто

виділити автоматичну фіксацію листувань за доступними каналами зв’язку, а також відображення активності кожного учасника на загальній панелі. Крім того, примітний інструмент Pipeline, що дозволяє користувачеві в пару кліків додавати успішні (або ні) продажі і наочно бачити зростання компанії, її рух до цілей.

Zoho CRM – величезний перелік інструментів, заходів, інноваційних технологій, які були створені для досягнення головної мети: утворити, розвинути і зміцнити будь-які взаємини з клієнтами, на рис 2.6 зображено головну сторінку. Zoho CRM дає можливість клієнтам займатися тим, що ті вміють найкраще: продажами. Zoho автоматизує робочі процеси.

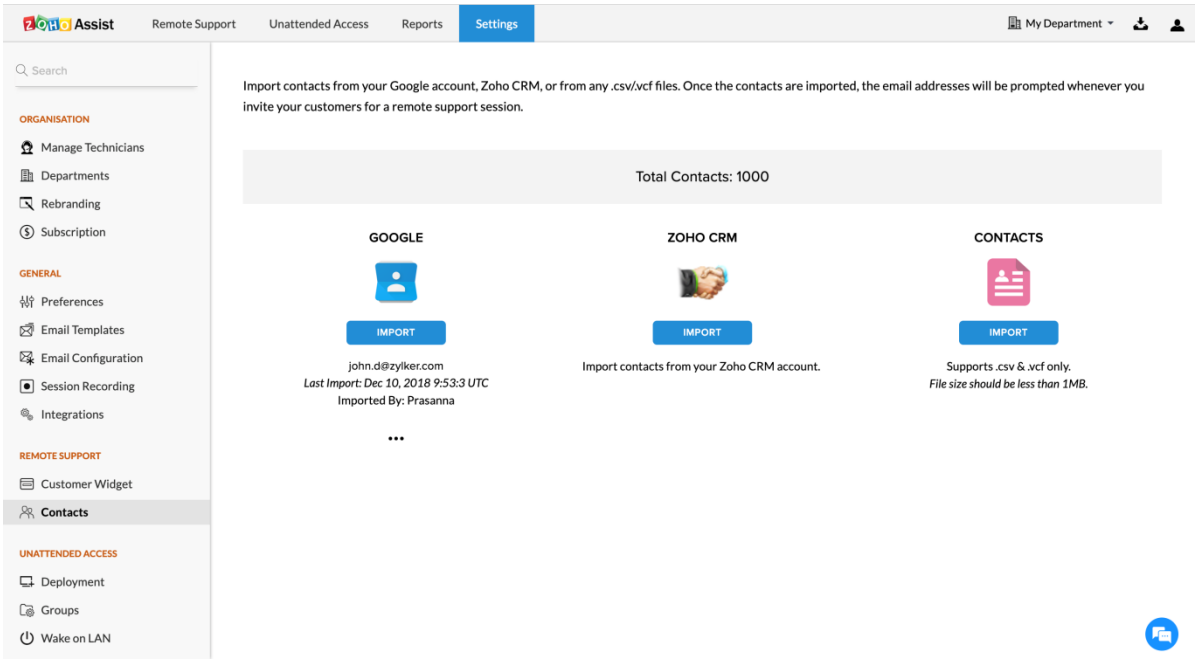

Рисунок 2.6 – Головна сторінка Zoho CRM

Робота системи Zoho CRM відбувається тільки через web-браузер, тому абсолютно ніяких програм на комп’ютер користувача не потрібно встановлювати.

Для початку слід придбати ліцензію. Ліцензія потрібна для того щоб отримати повний доступ до системи на місяць або ж рік. Будь-яка ліцензія надає користувачеві право підключатися за логіном і паролем безліч разів, наприклад, з ПК, лептопа і телефону. Одна ліцензія завжди відповідає тільки одному користувачеві в CRM. При цьому є можливість запустити будь-яку кількість одночасних сесій під одним логіном. Це потрібно для того щоб визначити число ліцензій, необхідних для придбання.

Тарифи системи Zoho CRM пропонує для користувачів 4 різних тарифи:

- Standart;
- Professional;
- Enterprise;
- CRM Plus.

Від конкретного обраного тарифу залежатиме функціонал CRM системи. Експерти вважають Enterprise найкращим тарифом, де є всі необхідні функції для середнього та малого бізнесу. Ціна однієї ліцензії Enterprise досить прийнятна і в даний момент становить близько 35 американських доларів на місяць на користувача.

У цій версії є будь-які функції Standart і Professional, а також додаткові опції, без яких важко уявити роботу CRM. При впровадженні CRM важливим моментом є можливість інтеграції з телефоном. І лише з Enterprise версією подібна інтеграція стала можливою. А ось у версіях Standart і Professional така інтеграція не була передбачена.

Після вибору типу ліцензії слід зрозуміти, які основні складові та можливості має Zoho CRM. Нижче розглянемо найбільш значущі з них при роботі з цією системою.

Складові системи поділяються на адміністратора та користувача. Користувачі – це ті, хто має доступ до системи. Володіючи унікальним логіном і паролем, користувач має можливість входити в Zoho CRM нескінченну кількість разів з різних гаджетів.

Адміністратор в свою чергу мають повний доступ до Zoho CRM і має можливість налаштовувати права доступу, а ще здійснювати контроль над іншими користувачами, займатися налаштуванням модулів і зовнішнього вигляду всієї системи, підключати різні плагіни і сторонні системи. Права адміністратора можуть бути у користувача, який стане займатися налаштуванням системи. При цьому в Zoho CRM відсутня ймовірність втратити доступ.

У інших користувачів можуть бути або не бути обмеження по роботі з системою. Слід зрозуміти, що до користувачів прив’язані будь-які вчинені ними дії,

що вкрай зручно для фіксації і аналізу інформації, прогнозування будь-яких процесів, формування звітів по працівникам і виявлення ефективності їх роботи.

Модулі – це сутність, крізь яку і здійснюється вся робота в Zoho CRM. Будь-які модулі представляються у вигляді вкладок, які завжди можна налаштувати, приклаад модуля для відображення графіків приведено на рис. 2.7.

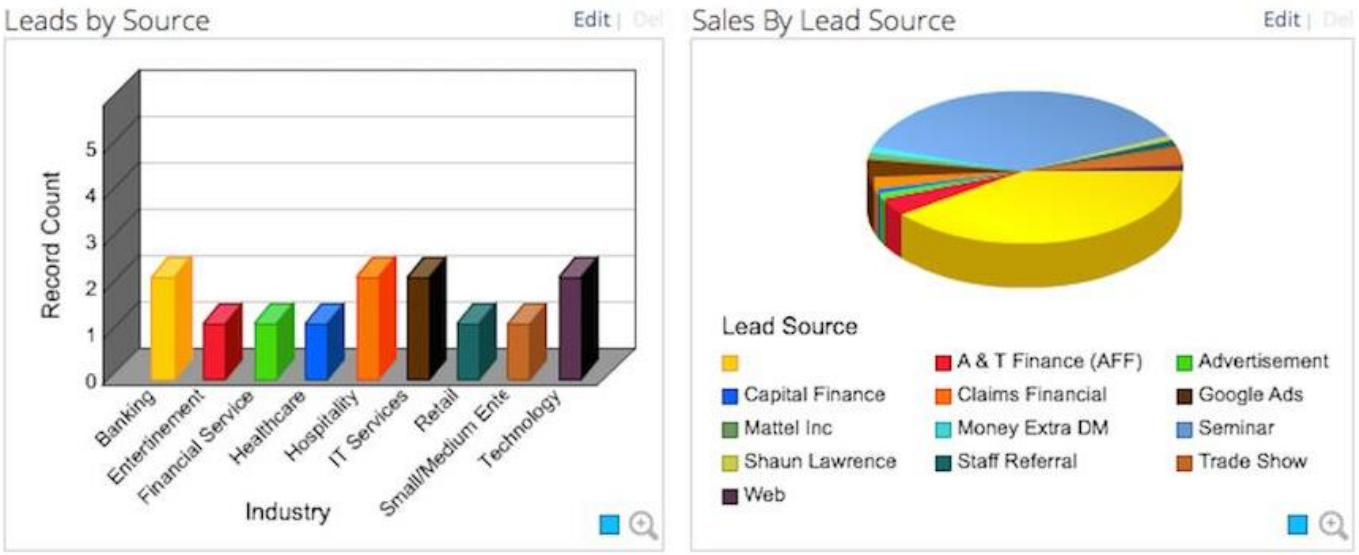

Рисунок 2.7 – Модуль для відображення графіків

У Zoho CRM є 2 різновиди модулів:

- попередні (або основні) модулі;
- модулі призначені для користувача.

Попередні модулі поставлені системою, які мають власну логіку роботи. Їх функціонал обмежений: певні сторони в них не піддаються змінам. Модулі призначені для користувача – це модулі створює користувач, і їх можна вільно змінювати. Будь-які модуль можна створити в додатку Zoho Creator. Також дана система має 3 базових модуля.

Лід – це потенційний клієнт, якимось чином зацікавився компанією. Це початкова стадія роботи. У контакт потенційного клієнта заносити ще рано. Ліди можуть бути генеровані з найрізноманітніших джерел. Список джерел різноманітний: виставки, email-кампанії, web-форми, обдзвони людей, маркетинг і багато іншого. Ліди можна створювати вручну або ж автоматично.

Контакт – людина, з яким взаємодіє користувач. Контакт – стрижень всієї системи, що відповідає на будь-які взаємодії. Це джерело сили будь-якого бізнесу. Чим більше буде надано інформації про клієнта, тим більше удачі і прибутку очікує бізнес. Це дуже потужний, персоналізований метод зберегти потрібні контактні дані в одному вікні. У модулі «Контакт» є все, що потрібно знати про клієнта:

- номери телефонів;
- електронні листи;
- завдання;
- нагадування;
- перелік справ;
- документи і інше.

Контрагент – компанія, пов’язана з конкретним клієнтом. Одному контрагенту може бути підпорядковане одночасно кілька контактів. Найчастіше робота з клієнтами проводиться в модулі Контакти, але модуль Контрагенти необхідний для їх угруповання. У такий модуль також входять налаштовані поля, але тут вже буде міститися юридична інформація про контакт.

2.3 Висновок

В даному розділі було розглянуто основні поняття CRM систем , які тенденції призвели до їх виникнення, яка мета цих систем та розглянуто основні типи, а саме операційні, аналітичні та спільні. Також були розглянуті основні термін які використовуються в роботі з CRM системами. Були описані деякі з представників CRM система, а сам Bitrix24, Hubspot CRM та Zoho CRM. Для більш зручного ознайомлення з недоліками та перевагами була побудована таблиця 2.1.

Таблиця 2.1 – Порівняння існуючих рішень.

Характеристика	Bitrix24	Hubspot CRM	Zoho CRM
Безкоштовність	+-	+	+-
Інтеграція з	+	-	+

іншими сервісами			
Багатоплатформеність	+	-	+
Велика кількість модулів	+	-	+
Можливість створення нових модулів	-	-	+
Зрозумілість інтерфейсу	+-	+	+

3 ОПИС АРХІТЕКТУРНОЇ РЕАЛІЗАЦІЇ ТА ЗАСОБІВ РОЗРОБКИ

3.1 Компонентний підхід [9]

Для повноти роз’яснення кожної окремої підсистема програмного забезпечення, необхідно розглянути кожен окремий компонентов, та його складові частини, також необхідно детально описати принцип їх використання згідно стандартних нотацій. Однак, кожен окремий компонент системи може наслідуватись від більше масштабного та значущого, з цієї причини, необхідно описувати не лише обрану функціональну частину програмного забезпечення, що є метою виконання магістерської дисертації, а й згрупувати їх за принципом наслідування. Докладний опис головних елементів буде розглянуто окремо.

Одним з важливих елементів системи є певна сутність, що відповідає за взаємодію вузлів системи, передачу даних, та цілістність програмного забезпечення в цілому, вона зберігає стани системи та реалізує певні інтерфейси. Такі елементи зазвичай називають “компонентами”.

Компонент поділяють на:

- логічні чи фізичні;
- технологічно незалежним, технологічно залежними
  або технологічно-зв’язаним;
- крупногранульованими чи дрібногранульованим.

Архітектура об’єкта проектується під час реалізація моделі предметної області. Задля створення більш дрібних компонентів у системі, під час проектування програмного забезпечення обирається архітектура, що має можливість поділу на функціональні частини. Для створення архітектурної моделі предметної області необхідно розглянути наступні окремі елементи:

- користувацький інтерфейсу, що надає замовник для впровадження;
- окремих процесів створення компонентів;
- відпрацювання запитів на операції зміни станів системи;
- дослідження бізнес-процесів, що протікають у системі та їх оптимізація..

Наступна частина магістерської дисертація присвячена розгляду засобів автоматизації, що використовує система задля збірки та складання загальної моделі. Дослідження рівню автоматизації системи встановило, що найбільш вагомий внесок збирання інформації здійснюють наступні фактори: програмно-технічна повнота розробляємого програмного забезпечення, відсутність надлишковості коду, його швидкодія та застосування необхідних артефактів присущих обраній предметної області.

У загальному вигляді, використовують наступні варіанти створення моделей проектів:

1) мануальна, тобто ручна збірка компонентів додатку у повноцінну систему.

У такому випадку, компонент складається з:

- детального опису архітектури система та кожного окремого компоненту;
- впровадження парадігм предметно-орієнтованих мов програмування;
- загальні концепції проектування інтерфейсу користувача.

2) загальна концепція, яка полягає у використанні самих актуальних інструментів для реалізації кожного окремого компоненту:

- інструментарій пошуку та розгляду окремих компонентів,
- автоматизація певних процесів розробки системи завдяки використанню сторонніх бібліотек, фреймворків.

Складовими частинами проекту є:

- опису архітектури системи та компонентів з яких вона слкадається;
- аналіз предметно-орієнтованих мов програмування для реалізації заявленого функціоналу;
- загальні відомості про концепції проектування користувацького інтерфейсу;
- формування загальних відомостей про систему що розробляється, яка генерується автоматично та класифікує кожен окремий компонент.

3) автоматизований процес створення моделі програмного забезпечення, що побудоване на необхідном інструментарії та формується автоматично.

Створення програмного забезпечення може бути реалізовано з використанням лише одного запиту до системи, проте це можливо лише за умови, що воно не потребує класичних методів розробки.

Складовими такого типу проектування є наступні елементи:

- дослідження можливих компонентів та побудова архітектури для їх реалізації;
- впровадження та використання принципів ООП та мов, що його підтримують;
- встановлення концепцій створення інтерфейсів;
- автоматизованого процесу пошуку елементів архітектури;
- деталізація та відокремлення компонентів за функціональними вимогами;
- оптимізація процесів створення програмного забезпечення.

Приємним доповненням є можливість генерувати документацію до системи автоматично.

3.2 Допоміжні компоненти в реалізації fronted частини

В роботі використовуються такі допоміжні компоненти як Angular, Typescript та Angular material. Розглянемо кожний з них.

Angular [10] - фреймворк JavaScript, який допомагає розробникам створювати додатки. Бібліотека надає безліч фіч, які роблять прості реалізації складних завдань сучасних додатків, таких як прив’язка даних, маршрутизація і анімація.

Angular також представляє ряд конвенцій про підходи до розробки додатків. Це може бути дуже корисно для великих команд, які повинні працювати разом на одному проекті. Angular – це одна з небагатьох бібліотек JavaScript, які забезпечують великий style guide з великою кількістю наочних прикладів того, як ви можете писати свій код, використовуючи цей фреймворк.

Технічно можливо використовувати Angular де завгодно, але найкраще він працює в нестандартних додатках до даних.

Angular працює не тільки з формами. Розробники створили безліч ігор за допомогою Angular і такі божевільні речі, як додатки з доповненою реальністю. Однак, більшість туторіалів і документації по Angular містять інформацію про створення деяких form-based додатків.

Angular хороший в form-based додатках, підходить для великих і складних додатків. Angular – не найпростіший і не маленький фреймворк JavaScript. Angular також добре підходить для додатків, які повинні працювати в декількох середовищах розробки. Якщо програма має працювати на веб, а також на Windows або Mac, можна дотримуватися одного з численних туторіалів для запуску Angular-додатків з популярним Electron project.

Angular Core Team складається з великої кількості людей в усьому світі і зі спільноти Angular. При цьому велика частина розробок Angular день у день здійснюється співробітниками Google. Приблизно 20 співробітників Google входять в Angular Core Team і всі ТОП-розробники проекту Angular є співробітниками Google.

Слід зазначити, що, незважаючи на контроль Google над Angular, сам фреймворк як і раніше багато в чому залежить від зусиль спільноти. Понад дві тисячі людей внесли свій вклад в open-source репозиторій Angular, в загальному доступі є незліченні туторіали і guides, численні компанії пропонують навчання і набір інструментів для розробників.

Якщо контроль над проектом належить одній компанії, це непогано, так як знижує конфліктні питання при прийняття нестандартних рішень.

По-перше, слід зазначити, що TypeScript [12] – це строго типізована і компільована мова. Хоча на виході компілятор створює JavaScript, який потім виповнюється браузером. Однак на відміну від JavaScript, дана мова сильно зменшую кількість помилок які виникнути без типізації.

По-друге, TypeScript реалізує багато концепцій, які властиві об’єктно-орієнтованим мовам, як, наприклад, спадкування, поліморфізм, інкапсуляція і модифікатори доступу і так далі.

По-третє, потенціал TypeScript дозволяє швидше і простіше писати великі складні комплексні програми, відповідно їх легше підтримувати, розвивати, масштабувати і тестувати, ніж на стандартному JavaScript.

По-четверте, TypeScript розвивається як opensource-проект і, як і багато проектів, хоститься на гітхабі. Крім того, він є кросплатформним, а це значить, що для розробки ми можемо використовувати як Windows, так і MasOS або Linux.

У той же час TypeScript є частиною JavaScript, а це значить, що будь-яка програма на JS є програмою на TypeScript. В TS можна використовувати всі ті конструкції, які застосовуються в JS – ті ж оператори, умовні, циклічні конструкції. Більш того, код на TS компілюється в javascript.

Згенерований компілятором TypeScript, що генерує JS підтримується більшістю браузерів, так як орієнтується насамперед на стандарт ECMAScript. В додаткових налаштуваннях кожен користувач може задати свою версію ECMAScript.

Оскільки дана мова є OpenSource, то все його інструменти доступні для всіх бажаючих. Для роботи з TypeScript ми можемо використовувати як Windows, так і Linux і MacOS.

Сам компілятор TS можна встановити за допомогою команди менеджера пакетів npm, який використовується в Node.js:

Angular Material [12] складається з набору попередньо встановлених компонентів Angular. На відміну від Bootstrap, який надає компоненти, які ви можете використовувати будь-яким способом, Anglate Material прагне забезпечити розширений і послідовний інтерфейс. У той же час він дає можливість контролювати, як поводяться різні компоненти.

Також, як Angular, Angular Material значно змінився з моменту його початкового випуску. Щоб додати в проект Angular Material, ми будемо використовувати команду ng add. Ця команда додасть бібліотеку в існуючий проект, а також тему CSS в angular.json. Вона також додасть скрипти в index.html і оновить AppModule.

Ще одна функція в Angular - ng add, яка оновлює залежності npm при випуску нової версії. Вона також оновлює пакети RxJS і Material Design, щоб використовувати переваги нових функцій.

Angular Material складається з декількох елементів, які відносяться до наступних категорій:

- елементи управління форм;
- кнопки і індикатори;
- навігація і макет;
- спливаючі вікна і модулі;
- таблиці даних.

Можливо згенерувати стартові компоненти за допомогою команди ng update. Через цю команду доступні деякі схеми:

- навігація;
- панель інструментів;
- таблиця.

3.3 Висновок

В розділі було розглянуто підходи до проектування проекту, а саме компонентний підхід. Описані його недоліки та переваги. Був описаний такий веб-фреймворк як Angular та описані його можливості. Також зроблено розгляд основної мови на якій написана програмна частина, а саме TypeScript. Також був приведений приклад основної бібліотеки з компонентами яка гарно працює з формами та веб-фреймворк Angular, а саме Angular Materia.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Важливим етапом розробки програмного забезпечення є вибір базової архітектури. У даній роботі було обрано підхід при якому кожна нова сутність буде представлена у вигляді окремого класу.

4.1 Опис основної архітектури додатку

Використання такої технологія як Angular накладає певні обмеження на архітектуру додатку. Це означає, що в додатку мають бути присутні такі архітектурні рішення як, модулі, компоненти, сервіси, інтерфейси та моделі.

Кожен модуль є класом з додаванням до його назви приставки Module, в цій роботі представлено такі модулі як AppModule, CoreModule та ShareModule. Кожен з перерахованих модулів має в свій архітектурі щось відмінне. Наприклад CoreModule, містить в собі основні посилання на бібліотеку стилів Angular Material, налаштування форм React Form, інтерфейси які будуть реалізовані в інших класах та моделі. ShareModule, містить сервіси які можуть бути використанні іншими модулями чи окремими класами. Також це модуль містить базові критерії які використовуються для аналізу в теорії ігор. AppModule містить в своїй архітектурі основні компоненти системи, клас AppComponent, як є базовим для всього додатку. Також в даному модулі описані інші компоненти так як HomePageComponent або StatistPageComponent. На рис. 4.1 можна побачити приклад структури розміщення класів.

4.2 Основні класи

4.2.1 Клас HomeComponent

Цей клас в об’єктній моделі представляє сутність для головної сторінки додатку. Саме в ньому ініціалізуються інші класи для майбутньої роботи з додатком. Одним з головних методів представлених в класі є метод “generationMatrix”. Даний

метод аналізу об’єкт те перетворює його на біль простий масив для майбутньої роботи з ним.

4.2.2 Клас ConditionsCompoent

Даний клас представляє сутність критеріїв котрі будуть проаналізовані в майбутньому. Цей клас повертає об’єкт за назвами критеріїв. Для додавання цих критеріїв використовується веб-форма з специфічними можливостями.

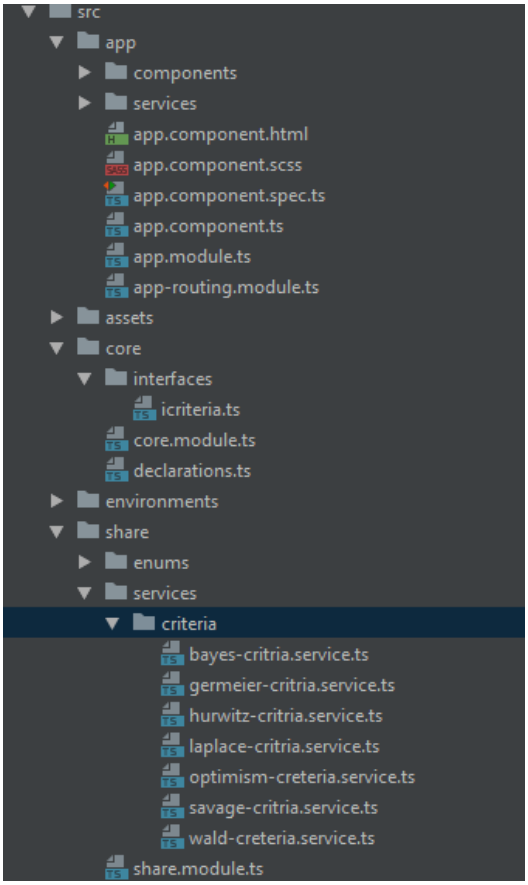

Рисунок 4.1 – Структура класів у межах модуля

4.2.3 Клас BayesCriteria

Цей клас повністю реалізує один із критеріїв теорії ігор, як зазначено в назві класу це — Критерії Байеса. Сам клас представлений з декількох методів, а саме “returnCriteriaResult”, метод який буде повертати результат за даним критерієм у вигляді масиву чисел. Наступний метод “countingByCriteria” реалізує вже сам

алгоритм підрахунку. В основі алгоритму є додавання усіх елементів строки, при множені кожного елементу на ймовірність. Ймовірність в огляді Критерію Байеса це одиниця поділена на кількість елементів в рядку. Після підрахунку суми рядку ми повинні отримати найбільше число з нами отриманих, це число і буде нашою стратегією. Дивитись у додаток А.

4.2.4 Клас GermeirCriteria

Цей клас реалізує критерій Гермейра. Як і клас, що описано вище клас має в сові два методи “returnCriteriaResult” та “countingByCriteria”. Однак алгоритм отримання стратегії відрізняється. Алгоритм представлений як ділення кожного числа рядка на ймовірність, де ймовірність це одиниця поділена на кількість чисел в рядку. Одна ділення відбувається якщо чисто буде більше нуля, в іншому випадку там треба помножити число на ймовірність. З отриманого результати з кожного рядку ми повинні виділити найменше число та порівняти їх. Число яке буде найбільшим і буде нашою стратегією. Дивитись у додаток А.

4.2.5 Клас HurwitzCriteria

Даний клас реалізує критерій Гурвіца. Як і клас, що описано вище клас має в сові два методи “returnCriteriaResult” та “countingByCriteria”, але на відміну від попередніх метод “returnCriteriaResult” отримує додатковий аргумент під назвою “pessimism”. Алгоритм даного критерію має в своїй основі використання додатковою зміною, а саме можливості песимізму. Для реалізації алгоритму нам потрібно знайти в кожному рядку найбільше та найменше число. Після знаходження двох чисел ми повинні перемножити мінімальне значення на значення песимізму та додати до нього максимальне значення перемножене на одиницю у якої відняли значення песимізму. Після даного підрахунку для кожного рядка ми повинні вибрати значення яке буде найбільшим, це число і буде нашою стратегією за критерієм Гурвіца. Дивитись у додаток А.

4.2.6 Клас LaplaceCriteria

Цей клас реалізує стратегію Лапласа. Стратегія Лапласа дуже схожа на стратегію Байеса. Клас як минулі реалізую два методи “returnCriteriaResult” та “countingByCriteria”. Оскільки критерій Лапласа схожий на критерій Байеса в їх алгоритмі є велика схожість. Однак в цьому критерії нам потрібно взяти кожне число з рядка та поділити на ймовірність, а після цього отримати суму нових чисел. З отриманих сум нам потрібно знайти найбільше суму. Стратегія з найбільшою сумою і буде нашою стратегією за даним критерієм. Дивитись у додаток А.

4.2.7 Клас OptimismCriteria

Цей клас реалізує критерій оптимізму. Як і класи, що описані вище цей клас також має два методи “returnCriteriaResult” та “countingByCriteria”. Алгоритм в свою чергу дуже простий і має в своїй основі знаходження найбільшого числа з рядка, а після його отримання треба знайти найбільше число у всій матриці. Номер рядка в якому буде це число і буде нашою стратегією. Дивитись у додаток А.

4.2.8 Клас SavageCriteria

Даний клас реалізує критерій Севіджа. Цей клас має такі методи “returnCriteriaResult”, “countingByCriteria”, “rotateMatrix” та “calculatingRisk”.

Метод “rotateMatrix” буде повертати матрицю на дев’яносто градусів. Метод “calculatingRisk” допоможе нам отримати найбільше значення кожного стовпця.

Алгоритм в свою чергу представлений у вигляді деякої послідовності. Першим кроком нам потрібно повернути матрицю на дев’яносто градусів та отримати з кожного рядка найбільше число за допомогою метода “calculatingRisk”. Другим кроком ми віднімаємо від найбільшого числа інші числа рядка та формуємо з ним нову матрицю. Третім кроком ми повертаємо матрицю в початкове положення

за допомогою метода “rotateMatrix”. Четвертим кроком ми отримуємо найбільше число з рядка та порівнюємо його з іншими найбільшими числами з кожного іншого ряду. Ряд в якому буде найменше число і буде нашою стратегією за критерієм Севіджа. Дивитись у додаток А.

4.2.9 Клас WaldCriteria

Цей клас реалізує критерій Вальда. Як і класи, що описані вище цей клас також має два методи “returnCriteriaResult” та “countingByCriteria”. Алгоритм цього критерію досить простий. З кожного ряду ми повинні отримати найменше число. Отримані числа ми порівнюємо між собою та знаходимо найбільше з них. Ряд в якому знаходиться дане число буде нашою стратегією. Дивитись у додаток А.

4.3 Висновок

У розділі було описано всі архітектурні рішення, що використано для побудови клієнтської частини, а також всієї системи в цілому. Був проведений огляд існуючих модулів таких як AppModule, ShareModule, CoreModule, також були розглянуті відмінності цих модулів. Були описані деякі компоненти та приведено приклад структурної схеми додатку. Були розглянуті основні типи класів окремо для кожного критеріїв та описано їх алгоритм роботи.

5. ІНТЕРФЕЙС КОРИСТУВАЧА

5.1 Основні частини інтерфейсу

Після відкриття додатку користувач попадає на головну сторінку, що зображена на рис 5.1. Дане вікно поділяється на 3 основних частини. Верхня частина відображає можливі переходи по додатку, в лівій частині користувач може обрати критерії аналізу.

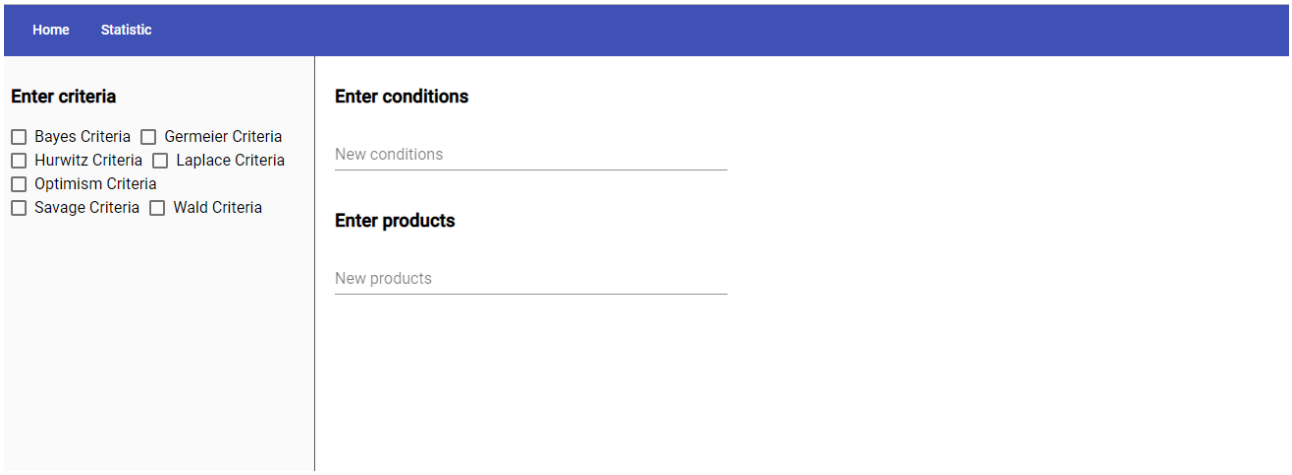

Рисунок 5.1 – Головна сторінка додатку

Для обрання одного з критерії критеріїв, треба звернутися для лівої частини екрану з текстом Enter criteria. Ліва частина екрану реалізована за допомогою окремого компонента SelectCriteriaComponent, розмітка даного компонента можна подивитися на рис. 5.2.

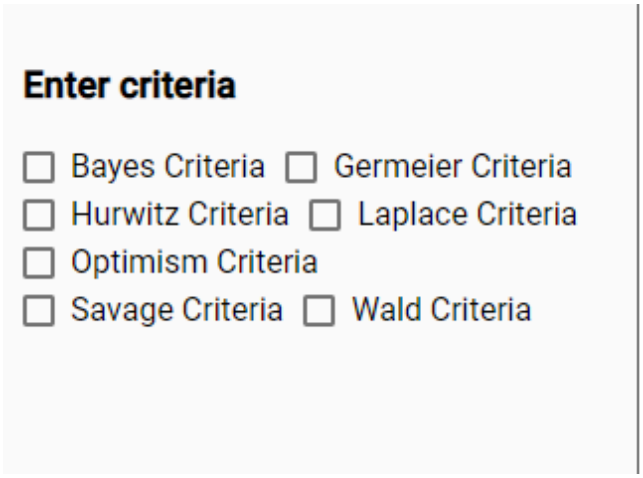

Рисунок 5.2 – Вибір критеріїв для аналізу

Одна для критерію Гурвіца потрібна додаткове поле, поле песимізму. Отже, якщо користувач захоче проводити аналіз за критерієм Гурвіца і натисне на нього, він побачить додаткове поле, поле можна побачити на рис. 5.3.

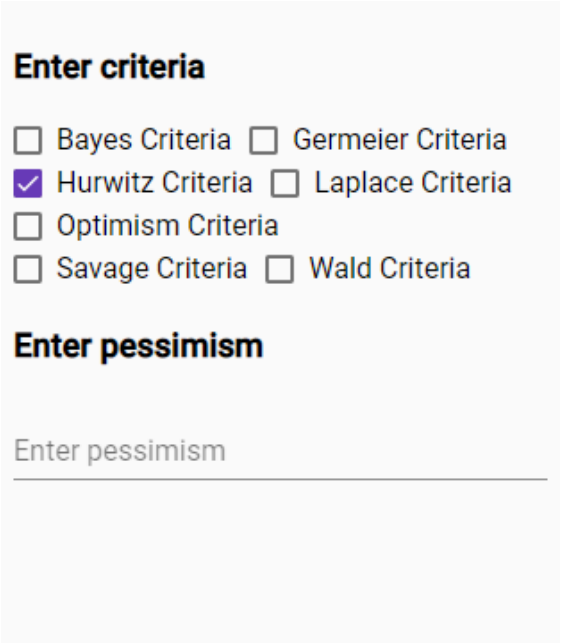

Рисунок 5.3 – Поле песимізму за критерієм Гурвіца

Для подальшого аналізу користувачу треба додати умови. Для цього додано окреме поле з заголовком Enter conditions. Ця розмітка реалізована за допомогою компоненту ConditionsCompoent. Для відображення було використано стилі бібліотеки Angular Material, а саме компонент ChipList. Цей компонент дає змогу додавати велику кількість умов для подальшого використання, даний компонент можна подивитися на рисунку 5.4.

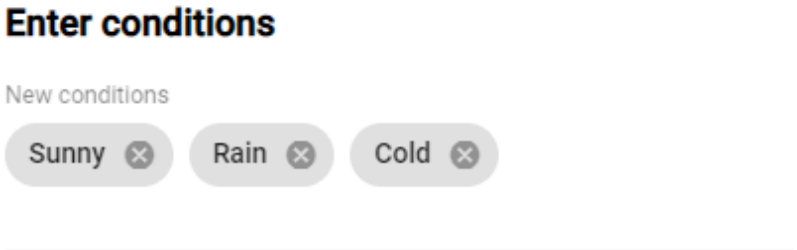

Рисунок 5.4 – Поле для додавання умов

Також в даному компоненті зроблена додаткова валідація на однакі назви умов. Якщо під час додавання буде знайдено однакову назву, компонент почне підсвічуватись червони, приклад зображено на рис. 5.5.

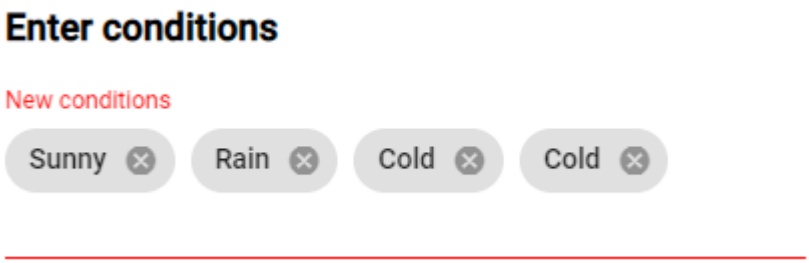

Рисунок 5.5 – Помилка при додаванні однакових умов

Для додавання елементів аналізу використовується цей самий компонент.

Для відображення матриці був створений компонент з назвою MatrixTable. Цей компонент використовує таблицю з бібліотеки стилів Angular Material. Також даний компонент використовую Reactive Form для швидкої валідації полів вводу. На рисунку 5.6 приведено приклад матриці.

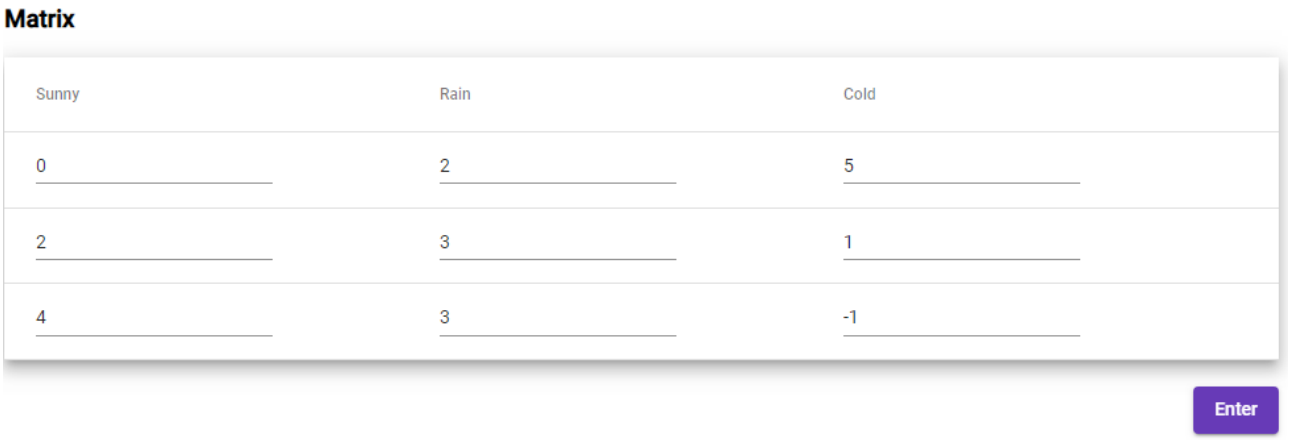

Рисунок 5.6 – Приклад матриці

Також слід зазначити, що підрахунки по матриці не можливі якщо хоча б одна с комірок буде порожньою. Приклад перевірки на валідацію на рис. 5.7.

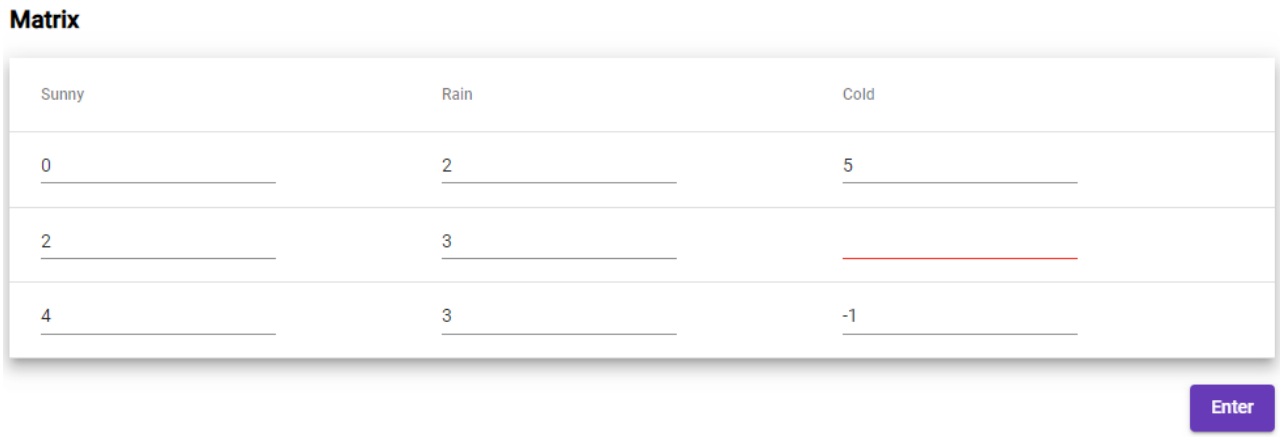

Рисунок 5.7 – Приклад валідації матриці

Для відображення результату аналізу була побудована таблиця, як один з найзручніших інструментів для сприйняття користувачем. Для даної таблиці також було використано бібліотеку стилів Angular Material. Приклад даної таблиці можна побачити на рис. 5.8.

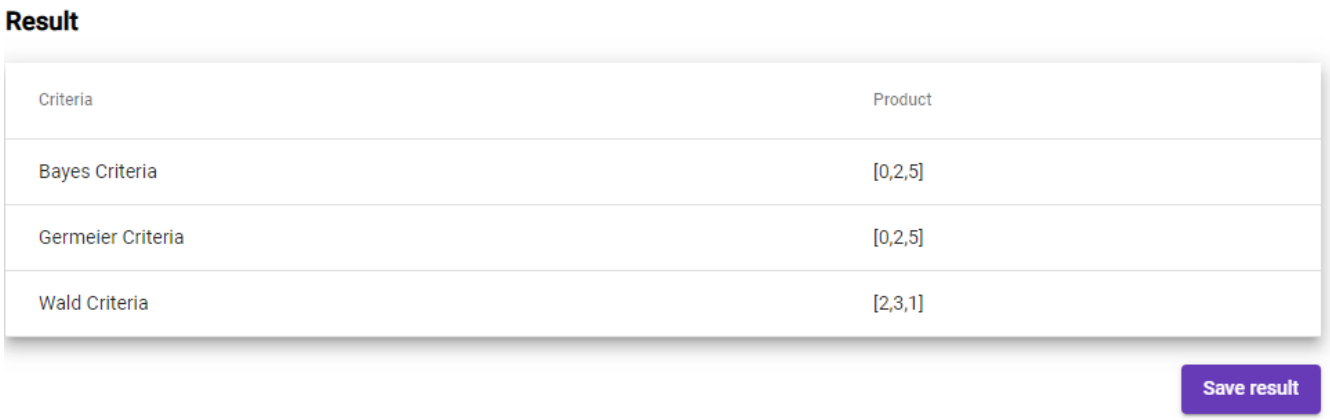

Рисунок 5.8 – Таблиця результатів аналізу

Якщо користувач хоче провести деякі порівняння він може натиснути на кнопку Save result, після натискання на котру користувач побачить спливаюче вікно, рис 5.9.

Дане вікно реалізується за допомогою компонента SaveModalComponent. В свою чергу вікно має в собі форму для вводу назви аналізу. Також поле вводу є обов’язковим для заповнення.

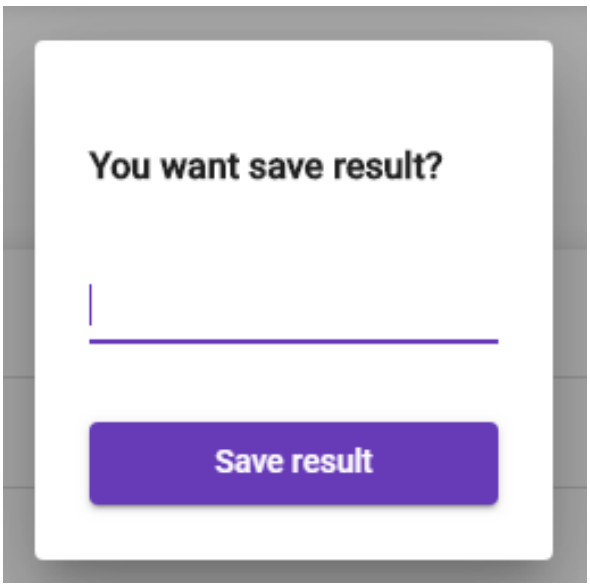

Рисунок 5.9 – Спливаюче вікно для збереження результатів аналізу

Якщо результати аналізу були збережені їх можливо продивитися на окремій сторінці. Для цього слід натиснути на кнопку в верхній чистині екрану з текстом Statistic. Якщо в даний момент результати будь яких аналізів будуть відсутні сторінка буде порожньою, рис. 5.10.

Рисунок 5.10 – Сторінка статистики з відсутніми даними

Якщо результати аналіз все таки присутні, користувач побачить спеціальний компонент з Angular Material [12], а сама Expansion Panel. Цей компонент дозволяє створити спеціальні частини розмітки як будуть відкриватися при натисканні на них, рис. 5.10.

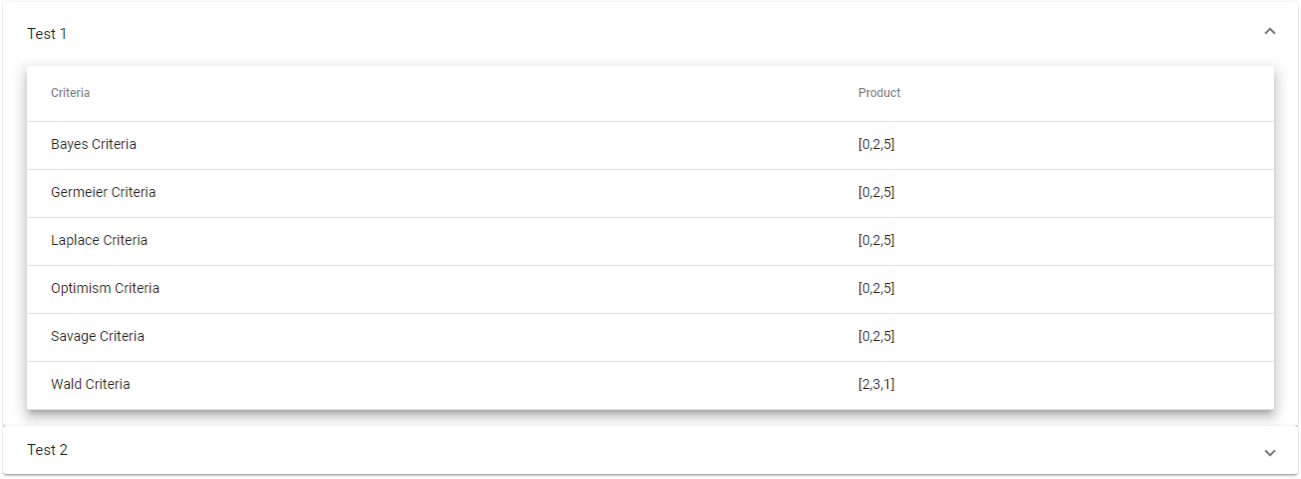

Рисунок 5.10 – Сторінка статистики з присутніми даними

5.2 Висновок

В даному розділі був проведений детальний аналіз інтерфейсу користувача. Описані компоненти з яких складається сторінка, також були описані компоненти які були запозичені з бібліотеки стилів Angular Material.

Було описано приклади валідації на сторінках та спливаючих вікнах. Був показаний приклад роботи з додатком та проведенням аналізу за допомогою нього.

6. ТЕСТУВАННЯ

6.1 Компонентне тестування [13]

Модульне тестування базується на перевірці кожної окремої функціональній частині проекту. Основним критерієм є повнота такого об’єкту. Модульними тестами, зазвичай, покривають цілі модулі, окремі класи або навіть функції реалізовані в системі. Для перевірки модульних тестів прийнято проводити процедури відпрацювання програмного коду при його виклику, такі функції, як правило, надає середовище розробки за допомогою фреймворків для тестування та комплексу інструментів для їх налагодження. Результати роботи тестів та знайдені баги прийнято виправляти в коді без збереження звітів про них з використання Bug Tracking System.

Серед усіх підходів до тестування, найкращі результати показує підхід, згідно якому тести реалізуються перш ніж пишеться програмна частина, такий підхід прийнято називати test-driven development або test first approach. Виходячи з назв типів модульного тестування стає зрозумілим, що першими кроками у створенні такого програмного продукту стає автоматичне генерування тестів, або їх написання згідно функціональних вимог до системи. Розглядаємий тип тестів відпрацьовує після реалізації кожної окремої частини, яку вони покривають. У загальному випадку, такі тести покривають дуже маленькі функціональні частини задля кращого результату. Розробка за даною методологією ведеться до тих пір, поки всі тести не відпрацюють корректно.

Розглядаючи переваги й недоліки компонентного і модульного тестування, можно зробити висновок, що в в загальному вони є майже ідентичними. Різниця між ними полягає лише у об’єкті тестування, для модульного тестування основною одиницею дослідження є конкретні значення, а для компонентного - функціональні частини системи.

6.2 Інтеграційне тестування [13]

Одним з найважливіших типів тестування у великих системах є інтеграційне тестування, метою якого є перевірка якості взаємодії та інтеграції модулів підсистеми у основну систему, а також якість взаємодії їх між собою та коректністю відпрацювання. Також, інтеграційні тести покривають ті компоненти системи, що пройшли тестування на більш дрібному рівні, наприклад: компоненти перевірені модульними тестами. Прийнято групувати результати модульних тестів у множини та перевіряти їх за стандартними планами тестування, після чого об'єднувати їх певним інтерфейсом.

У зв'язку з тим, що модулі пов'язані між собою інтерфейсами, що були створені під час реалізації програмного продукту, а інтеральні тести покривають цілі модулі системи не зважаючи, що відбувається у їх середині, можно назвати такий вид тестування - методом «чорного ящика». Якість результатів такого типу тестування залежить від якості модульних тестів.

Отже у зв'язку з тим, що інтеграційне тестування досліджує взаємодію між компонентами, а також їх зв'язком з іншими частинами системи, такими як: операційна система або іншими системи з якими взаємодіє модуль.

Інтеграційне тестування прийнято ділити на наступні рівні:

- компонентний інтеграційний рівень (Component Integration testing) - який тестує зв'язок компонентів системи, та відпрацьовує лише після проходження модульного та компонентного тестування;
- системний з інтеграційний рівнем - задачею якого є перевірка зв'язку між різними частинами системами, проведення цього типу тестування відбувається лише по завершенню системного тестування.

Виокремлюють три підходи до створення інтеграційних тестів:

- знизу вгору (Bottom Up Integration). Суть даного метода полягає у зборі всіх низькорівневих модулів системи, після чого всі вони проходять тестування. По завершенню тестування, інтеграційний тест переходить до наступного рівня і так просувається вгору. Для проведення такого типу тестування необхідно бути впевненим, що більшість необхідних модулів

вже реалізована, та коректно працює. За результатами даного тестування можно зробити висновки щодо загального стану розробки системи.

- зверху вниз (Top Down Integration). Принцип функціонування методу полягає у поступовому спуску по рівням системи, тобто першочергового система тестує найвищій рівень, після чого спускається на нижчі, усі нереалізовані компоненти системи замінюються заглушками з шаблонним функціоналом, та по мірі розробки системи їх замінюють на реальні функціонуючі модулі. Проте, таких підхід є не популярним.
- великий вибух ("Big Bang" Integration). Концепція даного виду інтеграційного тестування полягає у зборі всіх модулів системи у купу, та проходження усієї системи повного спектру впроваджених тестів. Даних підхід значно скорочує час на тестування, проте при виникнені помилок на нищчих рівнях тестування, таких як модульне, та інтеграційне тестування не дасть необхідного результату, та виникне потреба у проходження іншого типу тестів для виявлення баг тикетів, що призвили до помилок.

Для корректного відпрацювання інтеграційного тестування необхідно використовувати системи безперервної інтеграції. Принцип функціонування таких систем полягає у наступному:

- спочатку СБІ проводить ретельне дослідження систем контролю версій;
- у випадку виявлення змін у репозиторії, система стягує актуальну версію;
- після виконання системою попередніх пунктів, система проводить модульні тестування та перевіряє їх результати;
- по завершенню модульного тестування система компілює програмний код у вже відпрацьовані модулі;
- якщо всі низькорівневі тести відпрацювали коректно, система переходить до інтеграційного тестування;
- після отримання результатів тестування, система генерує звіт.

Таких механізм надає розробникам змогу проводити тестування кожен раз після оновлення версії програмного продукту та слідкувати за якістю роботи системи в цілому.

6.3 Системне тестування [13]

Основними тестами вже готового програмного продукту є системні тести. Такий вид тестування відповідає за відповідність функціоналу системи та її роботу взагалому згідно встановлених при проектуванні функціональних та нефункціональних вимог.

Системне тестування використовується для пошуку наступних видів дефектів системи:

- помилки у використання ресурсів;
- дефекти у цілісності даних;
- проблеми з сумісністю оточення;
- хибні методи взаємодії користувача з системою;
- відсутність певного функціоналу системи;
- погано реалізовані елементи системи, що робить їх використання не зручним для користувача.

Базується системне тестування на методі «Чорного ящика», який передбачає перевірку зовнішніх множин, які є обгорткою внутрішнього функціоналу системи. Використання такого типу тестування рекомендовано у різних оточеннях, задля встановлення проблем, які могуть виникнути у користувача. Необхідність такого тестування полягає у тому, що система, яка розробляється система має бути кроссплатформеною, та має підтримуватись в усіх доступних браузерах.

Виокремлюють декілька підходів до системного тестування:

- що базуються на певних вимогах. Тести створюються спираючись на функціональні та нефункціональні вимоги, що були розроблені на стадії проектування системи;

- що базуються на ймовірних випадків використання системи. Обраний підтип тестування будується на основі потреб користувача, та створюються прецеденти, що описують кожен окремий випадок, та проводять повний спектр тестування для виявлення дефектів у обраній середі.

Крім вище перерахованих підвидів тестування, можно відокремити альфа та бета-тестування, які використовують перед масовим запуском програмного продукту на ринок. Такий підхід мінімізує виникнення непередбачених наслідків у вже працюючої системи.

6.4 Приймальне тестування [13]

Приймальне тестування або Приймально-передавальне випробування (Acceptance Testing) – цей вид тестування проводиться для перевірки відповідності системи і використовується для :

- аналізу системи на відповідність приймальним критеріям;
- отримання відповіді від замовника або його представника чи задовольняє їх пророблена робота.

Дане тестування базується на вимогах та прописаних сценаріях використання додатку в майбутньому.

Щоб проводити дане тестування слід розглянути деякі пункти:

- якість продукту задовольняє сторону як замовника так і виконавця;
- представник замовника або сам замовник був посвідчений в план та сценарій проведення тестування, що був описаний ще на етапі підписання договору.

Тестування буде тривати до тих пір, поки якість продукту не буде задовольняти потребі замовника.

6.5 Результати тестування

У процесі тестування були перевірені основні частини функціоналу. Наступні 7 таблиць із нумерацією 6.1-6.7 містять інформацію про випробування та очікуваний результат.

Таблиця 6.1 – Процес додавання умов

Крок	Результат
Обираємо поле для додавання умови	Поле починає підсвічуватись знизу
Додаємо умову та натискаємо Enter	Умова додалась та була обведена сірим кольором

Таблиця 6.2 – Процес додавання однакових умов

Крок	Результат
Обираємо поле для додавання умови	Поле починає підсвічуватись знизу
Додаємо умову та натискаємо Enter	Умова додалась одна поле підсвічується червоним кольором

Таблиця 6.3 – Процес вибору критеріїв

Крок	Результат
Відмічаємо критерії	Поля зафарбовуються в синій колір
Обираємо критерій Гурвіца	Під списком критеріїв підмальовується додаткове поле

Таблиця 6.4 – Створення матриці

Крок	Результат
Заповнюємо поле умов трьома значеннями	Поля заповнилося та не відсвічується червоним
Заповнюємо поле продуктів трьом	Поля заповнилося та не відсвічується

значеннями	червоним, та відмалювалася таблиця з назвою матриця

Таблиця 6.5 – Обов’язкові поля під час заповнення матриці

Крок	Результат
Заповнюємо усі поля матриці крім останнього	Усі поля крім останнього заповнені
Натискаємо кнопку Enter	Поле яке не було заповнене відсвічується червоним кольором

Таблиця 6.6 – Підрахунок матриці

Крок	Результат
Обираємо критерії за якими буде проводитись аналіз	Отримуємо відмічені критерії
Заповнюємо поля умов та продуктів унікальними назвами	Відмалювалася таблиця з назвою матриця
Заповнюємо усі поля матриці та натискаємо кнопку Enter	З’являється таблиця з результатами аналізу за критеріями

Таблиця 6.7 – Збереження результатів аналізу

Крок	Результат
Натискаємо кнопку Save result	Відкрилося вікно з полем для назви аналізу
Заповнюємо поле	Поле не було підсвічено червоним кольором
Натискаємо кнопку Save result	Дані були збережені вікно закрилося

6.6 Висновок

У розділі було наведено короткі відомості про тестування програмного забезпечення. Були описані типи тестування, що використовувалися під час розробки додатку. Також були описані деякі випадки за крокам та очікуваннями. Оскільки більшість помилок було виявлено під час тестування можна стверджувати, що програма працює правильно.

7 СТАРТАП-ПРОЕКТ

Ідея яка буде інноваційною для ринку в даний проміжок часу називають стартапом. Одна не в кожен стартап буде вкладенно велика кількість часу, майнових та фізичних витрат. В першу чергу, це пов’язано з неможливість стартапу запропонувати щось нове або неможливістю зробити для нього велику піар компанію. Однак в наш час є інтернет з великою кількістю онлайн майданчиків для рекламування, розвитку та функціювання проекту.

7.1 Опис ідеї

Таблиця 7.1 — Опис ідею стартап-проекту

Зміст ідеї	Напрямки використання	Вигоди для користувача
	1. Вибір найкращої стратегії поведінки на ринку	Знаходження найкращої поведінки на основі вхідних даних
	2. Аналізування актуальності товарів підприємства	Заміна великих CRM систем на більш просту та зручну в розумінні

Основними характеристиками ідеї є:

1. кросплатформеність додатку;
2. доступ до системи онлайн;
3. можливості налаштування критеріїв аналізу;
4. доступ до теоретичних відомостей.

В якості програм для порівняння можна виділити Bitrix24 та Hubspot. Порівняння приведені в таблиці 7.2.

Таблиця 7.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту.

№	Техніко-економічні характеристики ідеї	Потенційні товари/концепції конкурентів			W (Слабка сторона)	N (Нейтральна сторона)	S (Сильна сторона)
		Мій проект	Bitrix24	Hubspot			
1.	Кросплатформеність додатку	Має	Має	Має	-	-	+
2.	Доступ до функціоналу системи онлайн	Має	Немає	Немає	+	-	-
3.	Можливість збереження системи інформації	Немає	Має	Має	-	-	+
4.	Багаті можливості налаштування процесу аналізу	Має	Має	Має	-	-	+
5.	Можливість додавання нових модулей	Має	Має	Має	-	+	-
6.	Доступ до теоретичних відомостей	Має	Має	Немає	-	-	+

7.2 Технологічний аудит проекту

Для розробки проекту, були проведені аналізи та обрані технології які допоможуть в швидкій реалізації.

Таблиця 7.3 — Технологічна ідея проекту.

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Збереження даних користувача	База даних Postgresql	Наявна	Доступна
2.	Реалізація бізнес-логіки та користувацького інтерфейсу	Веб-фреймворк Angular, мова програмування Typescript	Наявна	Доступна
3.	Серверна частина та зв'язок до неї	Google cloud, проткол HTTP	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: є можливою.				

Визначенні технології є безкоштовними та знаходяться у вільному доступі. Також дані технології є легкими для підтримки та майбутнього оновлення.

7.3 Аналіз ринкових можливостей запуску стартап-проекту

Для майбутнього запуску стартап-проекту на ринок, необхідно проаналізувати загрози, спланувати напрямки розвитку, визначити стан ринку, потреби споживачів та пропозиції конкурентів. Характеристика ринку в таблиці 7.4.

Таблиця 7.4 – Попередня характеристика ринку

№	Показники ринку (найменування)	Характеристика
1.	Кількість головних гравців	4 (підприємства, великий бізнес, малий бізнес)
2.	Загальний обсяг продаж, грн/ум. од.	Понад 1000
3.	Динаміка ринку (якісна ціна)	Зростає
4.	Наявність обмежень для входу	Відсутні
5.	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	100%

За результатами проведеного дослідження, ринок є привабливим, а норма рентабельності є задовільною.

Далі, необхідно визначити потенційних клієнтів. Характеристика описана в таблиці 7.5.

Таблиця 7.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Потреба в дешеве та зрозумілому в додатку, швидкість аналізу ринку	Підприємства, великий бізнес, малий бізнес	Використані при розробці технології, новий функціонал порівняний зі старим.	- легкість у використанні - низька ціна - якість аналізу

Після проведення аналізу стосовно потенційних клієнтів, було встановлено, що основна частина клієнтів це великі підприємства, великий та малий бізнес, тому важливо покрити як найбільше їх вимог. Після визначення вимог треба

проаналізувати фактори, що можуть завадити впровадженню проекту та визначити основні загрози (табл. 7.6, таблиця 7.7).

Таблиця 7.6 – Фактори загроз стартап-проекту

№	Фактор	Зміст загрози	Можлива реакція компанії
1.	Мала кількість функціоналу	Відсутність багатьох модулів які присутні в інших CRM системах	Використання сторонніх програм
2.	Вузьконаправленість	Система розрахована на одну конкретну задачу	Потреба в додаткових сервісах чи CRM системах
3.	Поява нової системи-конкурента на ринку	Поява конкурентів	Аналіз продукту конкурента, адаптація проекту

Таблиця 7.7 – Фактори можливостей стартап-проекту

№	Фактор	Зміст можливості	Можлива реакція компанії
1.	Швидкість в розумінні	Можливість швидко ознайомитись та інтегрувати на підприємстві	Використання як додатковою програмою разом з іншими аналітичними системами
2.	Масштабування	Збільшення можливого функціоналу за допомогою нових модулів	Масштабування програмного продукту

Для проекту був проведений аналіз основних загроз та можливостей. Більша частина загроз пов’язана тільки з малою кількістю функціоналу та вузькою направленістю проекту. Проте дані загрози не є дуже критичними, оскільки можуть бути виправленні в майбутньому з додаванням нових модулів. Таким чином проект може бути конкуренто спроможний, а значить треба провести аналіз конкуренції на ринку (табл. 7.8).

Таблиця 7.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив діяльності підприємства
Тип конкуренції: олігополія	Конкурентами є кілька великих систем	Залучення зацікавлених персон для покращення загальної якості
За рівнем конкурентної боротьби: національна	Ринок охоплює всі підприємства, великий. середній та малий бізнес	Забезпечення якості аналізу
За галузевою ознакою: внутрішньо-галузева	Конкуренти знаходяться в одній галузі (програмне забезпечення)	Аналіз факторів, які впливають на успіх у даній галузі
Конкуренція за видами товарів: товарно-видова	Товар конкурентів одного виду – програмне забезпечення	Проект орієнтований на великий. середній та малий бізнес
За характером конкурентних переваг: не цінова	Товар більш простий в ознайомленні та роботі з ним	Зосередження ресурсів на підтриманні вищої якості продукту
За інтенсивністю: не марочна	Товар тільки виходить на ринок	Забезпечити успішний старт продукту на ринку

Після ступеневого аналізу ринку, стає зрозуміло, що конкурентні продукти хоч і утримують ринок, але мають деякі недоліки та складності в експлуатації. Необхідний більш детальний аналіз умов конкуренції в галузі (табл. 7.9).

Таблиця 7.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Застарілі системи моделювання, Bitrix24, Hubspot	Будь яка програма з аналізу може бути конкурентом	Арендатори хостингових, хмарних сервісів, які використовуються в продукті	Підприємства, великий, середній та малий бізнес	Не беруч до уваги конкуренті товарів-замінникі на даний момент не знайдено
Висновки	Конкуренція існує і її інтенсивність не досить велика	Потенційних конкурентів достатньо	Існує невелика залежність від постачальників, проте вони не диктують умови роботи продукту.	Головна умова клієнтів точність аналізу та легкість в експлуатації	Таким чином, обмежень на ринку через товари-замінники не існує

Аналіз показав, що існують прямі конкуренти з більшою базою користувачів та функціоналу, але виникнення нових малоймовірно. Це означає, що проект може бути успішним, але для цього слід прибрати основні недоліки не втративши переваги. Обґрунтування факторів конкурентоспроможності описано в таблиці 7.10.

Таблиця 7.10 – Обґрунтування факторів конкурентоспроможності

	Фактор конкурентоспроможності	Обґрунтування факторів
1	Зручність використання	Продукт зменшує час на розуміння та

.		вивчення продукту
2.	Комплексний підхід	Поєднання усього функціоналу в одному місці покращує
3.	Можливість розширення продукту	Продукт передбачає можливість розширення додатку та гнучкість
4.	Точність підрахунків	Точність аналізу ринку при правильних вхідних даних

Для отримання максимального успіху на ринку були обрані фактори які є перевагами над конкурентами. Найважливішим з яких є легкість в освоєнні проекту. На основі визначених факторів конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 7.11).

Таблиця 7.11 – Порівняльний аналіз сильних і слабких сторін стартап-проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів порівняно з проектом						
		-3	-2	-1	0	+1	+2	+3
Зручність використання	6			Hubspot	Bitrix24			
Комплексний підхід	10					Bitrix24	Hubspot	
Можливість розширення продукту	5			Hubspot		Bitrix24,		
Легкість інтеграції на підприємствах	12		Hubspot		Bitrix24			

Порівняльний аналіз продемонстрував, що стартап-проект має переваги над конкурентами і може мати успіх на ринку. Для більш детального аналізу необхідно скласти SWOT-аналіз на основі ринкових загроз та можливостей (табл 7.12).

Таблиця 7.12 – SWOT-аналіз стартап-проекту

Сильні сторони: - новизна проекту; - легкість інтеграції на підприємстві; - легкість в розумінні та експлуатації.	Слабкі сторони: - несприятливе ринкове середовище; - вузьконаправленість.
Можливості: - розширення ринку в наслідок зацікавленості в продукті; - послаблення позицій конкурентів; - поява нових технологій.	Загрози: - небажання переходити на нову систему; - захоплення ринку новими сильними конкурентами.

Після аналізу були розроблені сприятлива поведінка на ринку (табл. 7.13). SWOT-аналіз дав змогу комплексно оцінити і порівняти ключові сторони проекту та ринкового середовища.

Таблиця 7.13 – Альтернативи ринкового впровадження

Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Загарбник	Значна	Максимум півроку
Наступник	Суттєва	Максимум рік

Для стартап-проекту були наступні ринкової поведінки – загарбник та наступник. Наступник підходить оскільки проект орієнтований на невеликий сегмент ринку, на самому ринку конкуренція представлена у вигляді олігополії. Використання даної альтернативи дозволить спеціалізуватися на одній частині

ринку, а потім поглиблювати свій вплив. Іншою альтернативою є загарбник, який орієнтований на захоплення ринку від конкурентів.

7.4 Розроблення ринкової стратегії проекту

Для досягнення поставленої мети на ринку, необхідно розробити стратегії поведінки, ка буде складатися з сукупності заходів, які спрямовані на отримання якомога більшого прибутку. Першим кроком буде визначення стратегії охоплення ринку (табл. 7.14).

Таблиця 7.14 – Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтований попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
Підприємства	Висока	Високий	Низька	Середня
Великий, середній та малий бізнес	Висока	Високий	Низька	Середня
Які цільові групи обрано: Підприємства і великий, середній та малий бізнес				

Серед потенційних користувачів були розглянуті підприємства, великий, середній та малий бізнес. В даному випадку кожний сегмент ринку є привабливим, оскільки проект не орієнтується лише на одну ланку ринку.

За результатами аналізу була визначена стратегія охоплення ринку. Оскільки потенційні споживачі можуть бути на різних ринках та можливостях фінансування. Наступний крок — це визначення базової стратегії розвитку (табл. 7.15).

Таблиця 7.15 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Наступник	Концентрація кожній ланці ринку	Надання клієнтам сучасного програмного продукту для аналізу поведінки на ринку	Стратегія масовий маркетинг

З базових стратегій розвитку була обрана стратегія масовий маркетинг, як найбільш вдала. Вона передбачає концентрацію на всьому ринку з представлення одного стандартизованого продукту. Наступним пунктом в розробці стратегії буде планування конкурентної поведінки (табл. 7.16).

Таблиця 7.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Схожі проекти вже існують	Спочатку проект буде витісняти існуючі проекти	Копіювання таких характеристик як: - легкість в експлуатації; - легкість інтеграції; - кросплатформеність продукту	Стратегія виклику лідера.

З існуючих стратегій була обрана стратегія виклику лідера. Стратегія виклику лідера полягає в активній конкуренції з лідером ринку з метою стати лідером самому. Наступним кроком буде визначення стратегії позиціонування (табл. 7.17).

Таблиця 7.17 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Зручний інтерфейс, точність	Стратегія масовий маркетинг	Стратегія виклику лідера	Можливість розширення системи, легкість експлуатації

Для аналізу стратегії позиціонування були розглянуті сильні сторони проекту, завдяки яким буде створена перевага перед конкурентами.

7.5 Розроблення маркетингової програми стартап-проекту

Наступним кроком розробки стартап-проекту буде маркетинг. Першим кроком якого буде формування маркетингової концепції. Для цього необхідно визначити ключеві переваги концепції потенційного товару (табл. 7.18).

Таблиця 7.18 — Визначення ключових переваг концепції і потенційного товару

Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
Потреба в легкому для	Легкість в освоєнні продукту з подальшим	Кросплатформеність додатку, легкість використання

освоєння програмному забезпеченні	розширенням за допомогою модулів	
Точність підрахунків	Зрозумілі алгоритми підрахунку на основі теорії ігор	Усім відомі алгоритми підрахунку з якими легко ознайомитись

Основною перевагою є зручність та зрозумілість роботи додатку в порівнянні з конкурентами. Також легка можливість інтеграції та розширення в майбутньому. Наступним етапом буде розробка трирівневої маркетингової моделі товару (табл. 7.19).

Таблиця 7.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
	Опис базової потреби споживача, яку задовольняє товар (згідно концепції), її основної функціональної вигоди		
	Надання сучасного програмного забезпечення для моделювання кодеків, що дозволить значно покращити навчальний процес		
	Властивості/ характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Економічні: зменшення ціни на підписку 2. Технологічні: використання сучасних технологій 3. Ергономічність: зручний користувацький інтерфейс, зрозумілий функціонал продукту 4. Естетичні: привабливий дизайн продукту	–/+	+/+/+/+/+

	Якість: стандарти та нормативи Міністерства Освіти України для навчальних закладів
	Документи виконані з логотипом підприємства
	Марка: AnalyticalGameTheory
	До продажу: представлення клієнтам продукту
	Після продажу: Підтримка продукту з віддаленого серверу
За рахунок чого потенційний товар буде захищено від копіювання: захищення інтелектуальної власності шляхом патентування	

Після формування маркетингової моделі треба визначити цінові діапазони (табл. 7.20).

Таблиця 7.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
500 ум. од.	500 ум. од.	900 ум. од.	50-250 ум. од.

Після аналізу аналогів були виставленні приблизні ціни на послуги. Таким чином товар є привабливим для ринку і більш доступним за конкурентів. Далі треба сформувати систему збуту (табл. 10.21).

Таблиця 7.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту

Замовлення проекту	Швидкість, виконання, надійність (надання консультацій та підтримка проекту)	Глибока	Власні сили

Для більш продуктивного збуту продукту було обрано збувати додаток власними силами. В цьому може допомогти велика кількість онлайн майданчиків. Останньою складової маркетингової програми є розроблення маркетингових комунікацій (табл. 7.22).

Таблиця 7.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Цільові клієнти весь час намагаються проаналізувати ринок та знайти найкращу стратегію	Інтернет, соціальні мережі і конференції	В Інтернеті і відповідних соціальних мережах	Донести інформацію про новий і більш зручний продукт	Рекламне звернення в доброзичливі формі

7.6 Висновки

В даному розділі було розглянуто потенціал виходу проекту на ринок з потенційними конкурентами. Проведено аналіз існуючих систем таких як Bitrix24 та Hubspot.

Були проведені аналізи техніко-економічного характеру для виявлення відмінностей від прямих конкурентів. Та виявлені деякі можливості комрціалізації проекту, а саме доступ до додатку онлайн, легкість в експлантації, приваблива ціна, можливість швидкого розвитку по ринку. З сукупності вище перерахованих факторі, можна зазначити, що продукт буде рентабельним для ринку.

Також були виявленні перспективи впровадження аналізуючи групи клієнтів які будуть зацікавлені, а саме підприємства, великий, середній та малий бізнес. Та виявлено, що основною проблемою для входження на ринок є не бажання великих гравців змінювати вже налаштовані системи. Одна посилаючись на велику складність цих систем та складності в їх розумінні та експлуатації конкуренція в сторону даного стартап-проекту стає більш привабливою та конкурентно спроможність проекту підвищується.

Подальша імплементація проекту має великий потенціал, адже проект новий і буде привносити до своєї системи тільки найкращі модулі конкурентів, але не буде втрачати власні переваги.

ВИСНОВКИ

У ході виконання магістерської дисертації була вирішена проблема економічного аналізу стратегії підприємств. Основними принципами є прозорість та точність аналізу при наявності певних вхідних параметрів.

Метою роботи є розробка системи, що може бути вільно інтегрована в будь який інший додаток, з повною автономністю від сторонньої системи.

Було проаналізовані схожі існуючі рішення, та встановленні їх переваги та недоліки. На основі аналізу була створена система, що представляє собою веб-додаток, що базуються на критеріях теорії ігор. Програмне забезпечення було реалізовано на основі технології Angular з використанням таких мов як TypeScript та JavaScript.

Після проведення аналізу та тестування методів, можна побачити, що алгоритм працює коректно та пропонує більш вдалу стратегію поведінки на ринку.

Завдяки цим дослідженням та розробці маємо можливість виходу на ринок з подальшим застосування у реальних системах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Василевич Л.Ф. Теорія ігор – КИИМ, 2000 – 76 с.

2. Петросян Л.А., Зенкевич Н.А., Сьоміна Є. А. Теорія ігор: Учеб. посібник для ун-тов - М .: Вища. шк., Книжковий будинок "Університет", 1998. – 304 с.

3. Теорія ігор. Дослідження операцій - [Електронний ресурс] – Режим доступ http://matica.org.ua/ – Дата доступу: 15.05.2012.

4. CRM система - [Електронний ресурс] – Режим доступ https://habr.com/crm/ – Дата доступу: 03.02.2015.

5. Белявцев М. І., Воробйов В. М., Кузнецов В. Г., Ншолайчук В. С Маркетинговий менеджмент: Навч. посібник / Під заг. ред. М 26 М. І. Белявцева та В. Н. Воробйова. — К.: Центр навчальної літератури, 2006. — 407 с.

6. Bitrix24: Офіційний сайт системи Bitrix24 [Електронний ресурс] – Режим доступу https://www.bitrix24.ua/ – Дата доступу: 04.02.2013.

7. Hubspot: Офіційний сайт системи Hubspot [Електронний ресурс] – Режим доступу https://www.hubspot.com/ – Дата доступу: 12.07.2017.

8. Zoho CRM: Офіційний сайт системи Zoho CRM [Електронний ресурс] – Режим доступу https://www.zoho.com/ – Дата доступу: 02.06.2014.

9. Лекція університету Київський національний університет будівництва та архітектури, лекції з дисципліни Архітектура програмного забезпечення – Режим доступу http://org2.knuba.edu.ua/ – Дата доступу: 16.04.2012.

10. Angular 2 Web Development with TypeScript, Sergey Barskiy – грудень 14, 2016 – ст. 34 – 47

11. Introduction to TypeScript, Kamran Ayub – січень 27, 2017 – ст. 59-64.

12. Angular Material: Офіційний сайт Angular Material [Електронний ресурс] – Режим доступу https://material.angular.io/ – Дата доступу: 15.07.2010.

13. Ревенчук І.А., Ткачова Т.С Якість та ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ – Харків: ХНУР 2016 – 43 с.

14. Горобчук Т. Мікроекономіка: Навчально-метод. посіб. для студ. економ. спец./ Тетяна Горобчук. – К.: ЦУЛ, 2002. –334 с.

15. Дегтерев Д. А. Теория игр и международные отношения / МИРОВАЯ ЭКОНОМИКА И МЕЖДУНАРОДНЫЕ ОТНОШЕНИЯ. – 2011, № 2. – С. 79-89.

16. Егоров А.И. Основы теории управления. – М.: Физматлит, 2004.

17. Жуковский В.И., Чикрий А.А. Линейно-квадратичные дифференциальные игры. – К.: Наукова думка, 1994.

20. Задоя А. Мікроекономіка: Курс лекцій та вправи: Навч. посібн./ Анатолій Задоя,. – К.: Знання, 2001. – 211 с.

21. Замков О.О., Толстопятенко А.В., Черемных Ю.Н. Математические методы в экономике. – М.: Изд. „Дело и сервис”, 2009.

22. Замков О.О., Толстопятенко А.В., Черемных Ю.Н. Математические методы в экономике. – М.: Изд. „Дело и сервис”, 2009.

23. Интриллигатор М. Математические методы оптимизации и экономическая теория. – М.: Прогресс, 2002.

24. Карагодова О. Мікроекономіка: Навч. посіб./ О. О. Карагодова, Д. М. Черваньов. – К.: Четверта хвиля, 1997. – 203 с.

25. Кілієвич О. Мікроекономіка для аналізу державної політики: Підручник/ Олександр Кілієвич, Олександр Мертенс. – К.: Вид-во Соломії Павличко "Основи", 2005. – 655 с.

ДОДАТКИ

Додаток А. Лістинг коду системи

Критерій Байеса

```
import {Injectable} from '@angular/core';
import {ICriteria} from '../../../core/interfaces/icriteria';

@Injectable()
export class BayesCriteriaService implements ICriteria {

  constructor() {
  }

  public returnCriteriaResult(matrix) {
    return this.countingByCriterion(matrix);
  }

  private countingByCriterion(matrix) {
    const bayesMatrix = matrix.map(array => {
      const length = 1 / array.length;
      return array.map(item => item * length);
    });
    const sumMatrixElement = this.returnArraySumMatrixElements(bayesMatrix);

    return matrix[sumMatrixElement.indexOf(Math.max.apply(null, sumMatrixElement))];
  }


  private returnArraySumMatrixElements(matrix) {
    return matrix.map(array => array.reduce((a, b) => (a + b)));
```

```
  }
}
```

Критерій Гермейра

```
import {Injectable} from '@angular/core';
import {ICriteria} from '../../../core/interfaces/icriteria';

@Injectable()
export class GermeierCriteriaService implements ICriteria {

  constructor() {
  }

  public returnCriteriaResult(matrix) {
    return this.countingByCriterion(matrix);
  }

  private countingByCriterion(matrix) {
    const germeierMatrix = matrix.map(array => {
      const length = 1 / array.length;
      return array.map(item => {
        if (item > 0) {
          return item / length;
        } else {
          return item * length;
        }
      });
    });
```

```
    const arrayMinValues = this.returnArrayWithMinValues(germeierMatrix);

    return matrix[arrayMinValues.indexOf(Math.max.apply(null, arrayMinValues))];
  }

  private returnArrayWithMinValues(matrix) {
    return matrix.map(array => Math.min.apply(null, array));
  }
}
```

Критерій Гурвіца

```
    import {Injectable} from '@angular/core';
import {ICriteria} from '../../../core/interfaces/icriteria';

@Injectable()
export class HurwitzCriteriaService implements ICriteria {

  constructor() {
  }

  public returnCriteriaResult(matrix, pessimism: number) {
    return this.countingByCriterion(matrix, pessimism);
  }

  private countingByCriterion(matrix, pessimism) {
    const hurwitzMatrix = matrix.map(array => {
      const maxValue = this.getMaxValue(array);
      const minValue = this.getMinValue(array);
      return this.calculatingCritiria(maxValue, minValue, pessimism);
```

```
    });

    return matrix[hurwitzMatrix.indexOf(Math.max.apply(null, hurwitzMatrix))];
  }

  private getMaxValue(array) {
    return Math.max.apply(null, array);
  }

  private getMinValue(array) {
    return Math.min.apply(null, array);
  }

  private calculatingCritiria(maxValue, minValue, pessimism) {
    return (pessimism * minValue) + ((1 - pessimism) * maxValue);
  }
}
```

Критерій Лапласа

```
import {Injectable} from '@angular/core';
import {ICriteria} from '../../../core/interfaces/icriteria';

@Injectable()
export class LaplaceCriteriaService implements ICriteria {

  constructor() {
  }
```

```
  public returnCriteriaResult(matrix) {
    return this.countingByCriterion(matrix);
  }

  private countingByCriterion(matrix) {
    const laplasMatrix = matrix.map((array) => {
      const length = array.length;
      return array.map(item => item / length);
    });
    const sumMatrixElement = this.returnArraySumMatrixElements(laplasMatrix);

    return matrix[sumMatrixElement.indexOf(Math.max.apply(null, sumMatrixElement))];
  }

  private returnArraySumMatrixElements(matrix) {
    return matrix.map(array => array.reduce((a, b) => (a + b)));
  }
}
```

Критерій оптимізму

```
import { Injectable } from '@angular/core';
import {ICriteria} from '../../../core/interfaces/icriteria';

@Injectable()
export class OptimismCriteriaService implements ICriteria {

  constructor() { }

  public returnCriteriaResult(matrix) {
```

```
    return this.countingByCriterion(matrix);
  }

  private countingByCriterion(matrix) {
    const optimismMatrix = matrix.map(array => {
      return this.getMaxValue(array);
    });

    return matrix[optimismMatrix.indexOf(Math.max.apply(null, optimismMatrix))];
  }

  private getMaxValue(array) {
    return Math.max.apply(null, array);
  }
}
```

Критерій Севіджа

```
import {Injectable} from '@angular/core';
import {ICriteria} from '../../../core/interfaces/icriteria';

@Injectable()
export class SavageCriteriaService implements ICriteria {

  constructor() {
  }

  public returnCriteriaResult(matrix) {
    return this.countingByCriterion(matrix);
  }
```

```
  private countingByCriterion(matrix: number[]) {
    const matrixAfterRotate = this.rotateMatrix(matrix);
    const afterCalculatingRisk = this.calculatingRisk(matrixAfterRotate);
    const startingMatrixPosition =
this.rotateMatrix(this.rotateMatrix(this.rotateMatrix(afterCalculatingRisk)));
    const arrayMaxValues = [];
    startingMatrixPosition.forEach(array => {
      arrayMaxValues.push(Math.max.apply(null, array));
    });

    return matrix[arrayMaxValues.indexOf(Math.min.apply(null, arrayMaxValues))];
  }

  private rotateMatrix(matrix) {
    const  result = [];
    for (let i = 0; i < matrix[0].length; i++) {
      const row = matrix.map(e => e[i]).reverse();
      result.push(row);
    }
    return result;
  }

  private calculatingRisk(matrix) {
    return matrix.map(array => {
      const maxValue = Math.max.apply(null, array);
      return array.map(value => maxValue - value);
    });
  }
}
```

Критерій Вальда

```
import { Injectable } from '@angular/core';
import {ICriteria} from '../../../core/interfaces/icriteria';

@Injectable()
export class WaldCriteriaService implements ICriteria {

  constructor() { }

  public returnCriteriaResult(matrix) {
    return this.countingByCriterion(matrix);
  }

  private countingByCriterion(matrix: number[]) {
    const arrayMinValues = [];
    matrix.forEach(array => {
      arrayMinValues.push(Math.min.apply(null, array));
    });

    return matrix[arrayMinValues.indexOf(Math.max.apply(null, arrayMinValues))];
  }
}
```